Graphical Calculi and their Conjecture Synthesis

Hector Miller-Bakewell

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2020

Abstract

Categorical Quantum Mechanics, and graphical calculi in particular, has proven to be an intuitive and powerful way to reason about quantum computing. This work continues the exploration of graphical calculi, inside and outside of the quantum computing setting, by investigating the algebraic structures with which we label diagrams. The initial aim for this was Conjecture Synthesis; the algorithmic process of creating theorems. To this process we introduce a generalisation step, which itself requires the ability to infer and then verify parameterised families of theorems. This thesis introduces such inference and verification frameworks, in doing so forging novel links between graphical calculi and fields such as Algebraic Geometry and Galois Theory. These frameworks inspired further research into the design of graphical calculi, and we introduce two important new calculi here. First is the calculus RING, which is initial among ring-based qubit graphical calculi, and in turn inspired the introduction and classification of phase homomorphism pairs also presented here. The second is the calculus ZQ, an edge-decorated calculus which naturally expresses arbitrary qubit rotations, eliminating the need for non-linear rules such as (EU) of ZX. It is expected that these results will be of use to those creating optimisation schemes and intermediate representations for quantum computing, to those creating new graphical calculi, and for those performing conjecture synthesis.

Contents

Chapter 1

Introduction

Whether or not researchers have achieved quantum supremacy is disputed [Aru+19] [IBM19], but it is undeniable that the power and complexity of quantum computers is increasing. In the words of Ref. [Pre18]: 'Now is a privileged time in the history of science and technology, as we are witnessing the opening of the NISQ era.' [1] The ability to run wider and deeper quantum circuits brings with it the desire to design more complicated quantum algorithms. Diagrams allow researchers to express these algorithms more clearly, and different types of diagram have different strengths. For an example see Figure 1.1, with the authors of that paper noting that 'the ZX calculus graph is more amenable to verification than the 3D diagram'. Graphical calculi go a step beyond simply being diagrams in that they come with rules for manipulating the graphical data directly.

Graphical calculi are not new. The ones we discus in this thesis flow, conceptually, from Roger Penrose's diagrammatic representations of tensor networks, but as Ref. [CK17] points out 'many similar diagrammatic languages were invented prior to this or reinvented later'. Here we shall be using these diagrams to consider Categorical Quantum Mechanics in the sense introduced in Ref. [AC09], a paradigm that lead to the introduction of what would become the ZX calculus [CD08]. In the words of Ref. [AC09]: 'The arguments for the benefits of a high-level, conceptual approach to designing and reasoning about quantum computational systems are just as compelling as for classical computation.' The ZX calculus has grown over time, in appearance, generators, scope and rules, with Figure 1.2 showing some of the changes in appearance. (The debate as to whether there is 'a' or 'the' ZX calculus is still ongoing[2], but we shall cover the different varieties in §2.2.) The story behind the development of the ZX calculus is probably best told by the titles of the various papers in Figure 1.3. In that list of titles we can see issues of 'incompleteness' highlighted, and then resolved, for different fragments of the calculus. With each challenge the calculus has grown, changed, and created off-shoots.

ZX, although probably the best known of the graphical calculi discussed here, is not

---

[1] NISQ devices are noisy, and have 'a number of qubits ranging from 50 to a few hundred'.
[2] Private communication during a ZX-calculus workshop

the only calculus built for qubit quantum computing. The completeness result 'A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics' [JPV18a] is constructed from the completeness result of another calculus called ZW [Had17]. ZX is built from the Z and X rotations of the Bloch Sphere; ZW is built from the W and GHZ states (it was originally called the GHZ/W calculus [Had17, p ii]). ZH, introduced in 2018 [BK19], is built from Z spiders and H-boxes. There is no need to stop here, and indeed this thesis introduces two new graphical calculi of its own. All these calculi are built to be powerful and intuitive ways of reasoning about a chosen domain. The utility, compared to standard matrix notation, is down to three factors: dimension, scale, and connectivity [Bac16, §2.1.2]. It is doubtless for these reasons that we see the use of graphical calculi spread not just into wider academia [BH20; GF19] but also into industry, with papers such as Refs. [BBW20] and [Cow+20] appearing with support from the company Cambridge Quantum Computing [CQC19].

Example 1.1 (Interchange Law [Mac13, p43]). The Interchange Law (given below) exemplifies the way that a graphical notation can convey meaning. Although we have yet to give precise meanings for the diagrams below we hope the reader can appreciate the way that the algebraic equation on the right is a tautology inherent in the notation on the left.

$$(a \otimes b) \circ (c \otimes d) = (a \circ c) \otimes (b \circ d) \qquad (1.1)$$

An analogy would be the way that the expression $f \circ g \circ h$ expresses the associativity of the composition by virtue of the lack of brackets.

There is a second strand to this thesis, and that is Conjecture Synthesis. Isabelle Conjecture Synthesis (IsaCoSy) was created with the '[...] aim to automatically produce a useful set of theorems' [JDB09]. What made IsaCoSy, expanded upon in Ref. [JDB11], different from previous theory formation systems is that of method. Previous examples had used a *deductive* approach (from the known facts make more facts) whereas IsaCoSy uses a *generative* approach (generate all hypotheses, weed out false ones). While this method is, on the face of it, intractable, the authors of Ref. [JDB09] realised that they only needed to generate hypotheses that didn't reduce to any other hypothesis. As a simple example the hypothesis '$x = 1 + 0$' is automatically redundant if you are also going to generate the hypothesis '$x = 1$'. We shall get into the graphical calculus version of conjecture synthesis in more detail in §3, but it was this idea that spurred the research presented in this thesis: 'Can we make conjecture synthesis work well for graphical calculi?'

The implicit first part of the question ('Can we make conjecture synthesis work *at all* for graphical calculi?') had been answered by Kissinger [Kis13] in a working algorithm written as a module for Quantomatic [KZ15]. Quantomatic then went through a rebirth
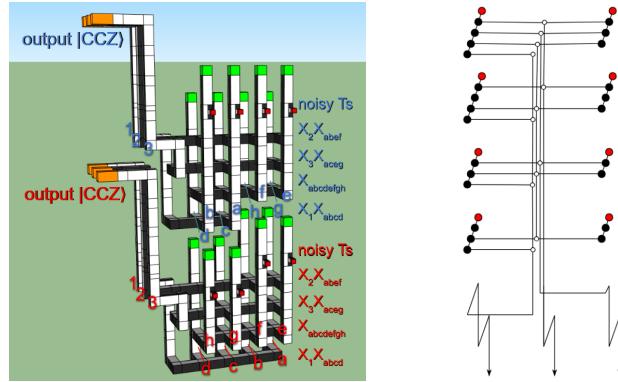
Figure 1.1: A representation of a CCZ factory in 3D (left) and (something similar to) the ZX calculus (right), as presented in Ref. [GF19, Figure 9].
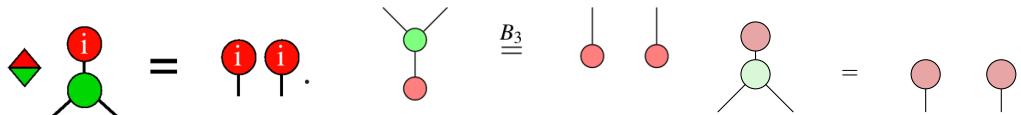
Figure 1.2: The stylistic change in the ZX calculus between 2008 [CD08] (left), 2011 [CE11] (middle), and this thesis (right). The colours used in this thesis have been chosen to be easier to distinguish between for anyone with red / green colour vision deficiency, or when printed in greyscale, with the green node appearing lighter than the red. [Mil19]

from ML into Scala and a Scala-based implementation of a similar algorithm was implemented by Kissinger and the author of this thesis. While this produced many results (over the course of the research for this thesis there were over 100,000 files produced, all with conjecture synthesis in mind) it did not produce, to the author's knowledge, any particularly *useful* results. The reason for this is that even after running the algorithm for two weeks on the group's computing cluster the algorithm was still only producing theorems that a researcher could have shown by hand, of at most four vertices per diagram.

The theorems produced by QuantoCoSy (as this program had now been named) often showed high levels of similarity between them. These were not redundancies like the aforementioned '$x = 1 + 0$' reducing to '$x = 1$'. Instead these similarities represented 'higher' structure inherent in the phase group or in repeated sections of the diagrams. The direction of the research then became 'how can we infer and verify conjectures in this higher structure?'

This thesis comes in two parts. In the first we generalise on, investigate, and create new graphical calculi, exhibiting new ways of comparing and relating these calculi, and examining diagrammatic maps that preserve equational soundness. In the second we discover a deep link between phase algebras and algebraic geometry, allowing us to infer and then verify parameterised equations. Perhaps the clearest example of these strands coming together is in §8.3, where we use what we have learned about soundness-preserving maps, the geometric structure of parameter spaces, and some Galois Theory, to show that a ZX, ZW, ZH, or RING diagram with phase variables can be verified by a *single* equation without phase variables. This single, simple equation implies and is implied by the large, parameterised family of equations. Our choice of phase algebra directly dictates the geometry of the parameter space, which in turn dictates the generalisations of our theorems.

As quantum computing algorithms get steadily more complicated, and as the diagrams get larger we shall need to be able to automatically produce the useful set of theorems that conjecture synthesis promises us. We shall need a toolbox of useful theorems for proof assistants (like Quantomatic or PyZX [KW20a]), or for intermediate representations and optimisers of quantum algorithms [Mur+19; Cow+20; Bac+20]. Our researchers will also need new tools and new calculi. This thesis provides advances in all of these areas. The more efficiently we can reason about quantum computing the more efficient we can make our circuits, and the faster we shall achieve the world where quantum computing is directly helpful.

Reference	Published	Title
[Sch13]	2013	The ZX calculus is incomplete for non-stabilizer quantum mechanics
[Bac14a]	2014	The ZX-calculus is complete for stabilizer quantum mechanics
[DP14]	2014	Pivoting makes the ZX-calculus complete for real stabilizers
[SZ14]	2014	The ZX-calculus is incomplete for quantum mechanics
[Bac14b]	2014	The ZX-calculus is complete for the single-qubit Clifford+T group
[Bac15]	2015	Making the stabilizer ZX-calculus complete for scalars
[BPW17]	2017	A Simplified Stabilizer ZX-calculus
[Jea+17]	2017	ZX-Calculus: Cyclotomic Supplementarity and Incompleteness for Clifford+T Quantum Mechanics
[JPV18a]	2018	A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics
[NW17]	2017	A universal completion of the ZX-calculus

Figure 1.3: A table detailing the path towards the completeness results for several fragments of the ZX calculus. The year shown is the year of publication, but it should be noted that the work of [NW17] builds on the work of [JPV18a], despite being published beforehand.

1.1   The contributions of this thesis

The author hopes that the reader finds the entirety of this thesis useful, but this section exists to draw attention to the most likely candidates of interest, grouped loosely by theme. The main results chapters come after two background chapters: §2 for graphical calculi, and §3 for conjecture synthesis. The author would like to highlight the applications of Algebraic Geometry and Galois Theory as particularly novel to the study of graphical calculi or quantum circuits, as well as the construction of the two new calculi RING and ZQ.

1.1.1   The calculus RING

RING is a universal, sound, complete graphical calculus built just from ring operations and compact closure. As such it acts like a unifier: Any graphical calculus with a similar phase ring structure contains a copy of RING. Examples of such calculi are ZW and ZH. RING and ZX form the bulk of the examples used in this thesis.

> ▷ The calculus RING_R is introduced and shown to be complete (§4)

> ▷ We justify referring to the calculus $\text{RING}_{\mathbb{C}}$ as 'the' generic phase ring quantum graphical calculus (Remark 5.36)

1.1.2   Parameterised rules as geometric surfaces

We demonstrate how to present parameterised rules as geometric surfaces, preserving existing ideas of rule linearity from the literature and introducing an algebraic geometric approach to graphical calculi and conjecture inference.

> ▷ We provide a geometric framework for parameterised rules (§7.2.1)

> ▷ We demonstrate how to convert linear rules to this geometric framework and back again (§7.2)

> ▷ We use this link to infer the existence of linear rules for conjecture synthesis (§7.2.4)

1.1.3   Phase homomorphism pairs

Phase homomorphisms are maps that act on the phases of diagrams. We define here phase homomorphism *pairs*; phase homomorphisms paired with an additional functor that then commutes with the interpretation. An example of a phase homomorphism pair negates phases in the ZX calculus, and in doing so creates the complex conjugate of the corresponding matrix. In this thesis we codify, explore, and classify this notion, showing that phase homomorphism pairs preserve soundness, and even preserve proofs, in some of the calculi we consider.

▷ We show that the space of equational theorems is closed under the action of phase homomorphisms for RING, ZW, and ZH (Theorem 5.42)

▷ We classify the phase homomorphism pairs for RING, ZW, ZH, and the finite fragments of ZX containing $\pi/4$ (§5.4)

▷ We link phase homomorphism pairs to geometric symmetries of parameter spaces (§7.2.5)

1.1.4   !-box verification

We introduce a method for verifying equations containing !-boxes by verifying a finite number of equations without !-boxes. This verification depends on a property we call *separability*. This is possible thanks to a new way of presenting successive !-box iterations in a manner that depends on $\circ$ rather than $\otimes$ products.

▷ We define the notion of separability in §8.4.1

▷ Series !-box form is defined in §8.4.2

▷ Finite verification of !-boxes is shown in Theorem 8.51

1.1.5   Phase variable verification

We show how to verify families of equations parameterised by phase variables using a method built on polynomial interpolation. This method requires checking a finite number of sample values for the variables.

▷ Phase variables and !-boxes can be verified in a finite manner without reference to either (Theorem 8.58)

1.1.6   Applications of Galois Theory

For certain languages we can encode automorphisms of field extensions as phase homomorphism pairs. In doing so we can use the machinery of Galois Theory to manipulate certain phases in a quantum circuit while preserving others. This is combined with the symmetries provided by phase homomorphism pairs to further reduce the number of equations needed for verification.

▷ Over qubits we can verify all the phase variables in a diagram using a single equation containing no phase variables (§8.3)

1.1.7   The generalisation step

Conjecture synthesis saw a recent advancement with the generative model discussed in §3. In this thesis we introduce an new step to this process; generalisation of theorems at the point of synthesis. This step requires a conjecture inference framework such as the one introduced in this thesis in order to be possible.

> ▷ We alter the generative conjecture synthesis method with the addition of a generalisation step (Remark 3.3)

1.1.8   The graphical calculus ZQ

The foundation of the calculus ZX is the Z and X spiders that represent rotations of the Bloch Sphere. These rotations generate all possible rotations of the sphere via Euler angles. Rather than take a generating subset of these rotations as vertex-labels, ZQ allows arbitrary rotations (expressed as quaternions) as edge-labels. This leads to a calculus that the author feels is clearer and more intuitive than ZX.

> ▷ The calculus ZQ is introduced and shown to be complete (§6)

> ▷ The spiders of ZX are shown to not be viable for representing a non-commutative group (Corollary 6.11)

> ▷ ZQ is the first qubit graphical calculus to use a non-commutative phase group

> ▷ The rules of ZQ are linear, whereas the rules of ZX are not (Remark 7.31)

1.1.9   Generalising ZH

The calculus ZH treats phase-free Z spiders and generalised Hadamard nodes as fundamental building blocks [BK19]. It was announced in 2018 and was shown to be complete via the construction of a normal form. For the rest of this thesis we shall refer to this version as $ZH_{\mathbb{C}}$, because the phases of these generalised Hadamard nodes are elements of $\mathbb{C}$. It can be shown, however, that by simply changing this parameterisation to any commutative ring with a half, R, that a complete and universal calculus is then exhibited for R-bits.

> ▷ The calculus ZH_R is generalised from $ZH_{\mathbb{C}}$ and shown to be complete (§4.4)

1.2   Overview

The eventual conclusion of this thesis is that phase algebras are a powerful and important aspect of a graphical calculus, allowing us to apply techniques and knowledge from previously unrelated disciplines to the field of quantum graphical calculi. We will use

this knowledge, among other new results, to also further the field of conjecture synthesis with the introduction of a new 'generalisation' step. Along the way we will introduce new calculi, novel links to algebraic geometry, and novel inference and verification methods.

Following the two background chapters we will introduce the complete, universal calculus RING in §4, which will serve as an important example in later chapters. §5 will then discuss the place of RING in the wider context of phase-ring graphical calculi, and the ways in which phase algebra homomorphisms interact with diagram equations.

The thesis will then explore another new calculus in §6, explaining the necessity behind its novel presentation, and the way in which it solves the issues with the (EU) rule of ZX. §7 establishes the novel link between parameter spaces and geometric surfaces, building on the phase homomorphism pairs from §5. We also cover why this framework works for ZQ and not ZX. This link forms the core of our conjecture inference and is directly linked to the verification methods of §8. §8 introduces new verification results and then brings these together with the phase homomorphism pairs of §5 and some Galois Theory in what the author feels is quite a beautiful way. This allows the verification of an equation involving phase variables by verifying a single equation not involving phase variables. In §9 we conclude the thesis.

1.2.1   An Algebraic Preview

Given that quantum graphical calculi exist at the intersection of mathematics, physics, and computer science it is likely that some readers may have never encountered the topics of Algebraic Geometry, Galois Theory, or Laurent Polynomials. Since we mention these ideas before giving a proper treatment of them an overview may be appreciated. In their broadest terms:

- Algebraic Geometry is based on the idea that there is no difference between a geometric surface in some space, and the collection of polynomial equations that define that surface. For example the points of the circle described by $x^2 + y^2 - 1 = 0$ are still solutions to the equation $x^3 + xy^2 - x = 0$. This observation allows us to switch between considering collections of polynomials and the surface itself.

- Galois Theory concerns pairs of fields, one inside the other, and the homomorphisms of the larger field that act as the identity on the smaller. A clear example is complex conjugation acting on the fields $\mathbb{R}$ and $\mathbb{C}$: The real numbers are unaffected by complex conjugation, but the elements of $\mathbb{C} \setminus \mathbb{R}$ are moved. We will use Galois Theory to construct maps that fix certain phases in a diagram, but move others.

- Laurent Polynomials are a generalisation of traditional polynomials by allowing positive *and negative* powers of the indeterminants. Importantly they can still be added, multiplied, and factorised in the same manner as traditional polynomials. Once we introduce them we will be working to convert our equations involving Laurent polynomials into equations using traditional polynomials instead.

These topics are covered in far more detail when their uses arise in this thesis.

1.2.2   Ethical considerations

There is justified concern about the impact of quantum computing on humanity. The impact on classical cryptography is probably the most important [Sho94], but the author hopes that, as with classical computing, the net effect will be undoubtedly positive. An example of a near-term positive is advances in chemical synthesis and protein folding for medical research [Per+12].

1.2.3   Chapter dependencies

Each of the results chapters contain a short introduction and gives an indication of which earlier chapters they are dependent upon. We also include a dependency diagram in Figure 1.4. Although §6 is not strictly a dependency for §7 there are examples and remarks beyond the core results that will require knowledge of ZQ.
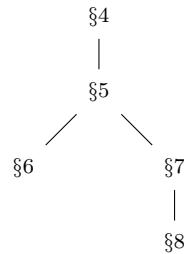
Figure 1.4: The dependency structure of the chapters in this thesis.

Before embarking on any new results we will spend the next two chapters covering the motivation and context for this thesis. The final note for this introduction is that unless stated otherwise all results were the work of the author.

Chapter 2

Graphical Calculi

In this chapter:

    ▷ We cover graphical calculi, graph rewriting and reduction systems

    ▷ We cover the calculi ZX_G, ZW_R, and $\mathrm{ZH}_\mathbb{C}$, among others

    ▷ We cover phase variables and !-boxes

Graphical calculi, in the context of quantum computing, stem from Penrose's diagrammatic notation for tensors [Pen71; CK17]. The benefit of graphical notation for tensors is that these diagrams are inherently two-dimensional, in contrast to the single dimension used for writing out algebraic terms. This allows us to use one dimension (horizontal composition) to represent $\otimes$ and the other (vertical composition) to represent $\circ$. Since Penrose's work the idea of tensors has been generalised to that of a monoidal category [JS91], and with this generalisation has come a host of graphical calculi [Sel11]. Graphical calculi are not just representations to help intuition, but are powerful tools for reasoning about monoidal categories (see the coherence theorems of Ref. [Sel11]).

We present the desiderata of a graphical calculus in Remark 2.1, with a concrete definition and many examples to follow. Importantly, graphical calculi need to be both graphical (i.e. based on diagrams) and calculi (i.e. can be used for calculation). The second part is the interesting one; we use the diagrams to represent some other object of study, and then manipulate the diagrams to represent manipulations of the object of study. The important properties of calculi, which will receive formal definitions in a moment, are *soundness* (the calculus can only show things that are true) and *completeness* (anything that is true and can be stated can also be shown). The 'and can be stated' part introduces a third property: A graphical calculus is *universal* with regards to some collection of objects if every one of those objects can be represented in the graphical calculus.

Almost certainly the best known example of a graphical calculus in quantum computing is simply that of quantum circuit diagrams (see Ref. [NC10]). Probably the best

known example of the calculi considered here is the ZX-calculus (Definition 2.16) which uses diagrams built from red and green vertices to represent matrices. It differs from quantum circuit diagrams not just in terms of generators, but also because ZX represents morphisms in a compact closed, dagger, symmetric monoidal category, compared to the 'plain' monoidal category of circuit diagrams. In a very real sense the power of ZX comes from this extra structure.

Remark 2.1 (Desiderata for Graphical Calculi). We desire the following aspects for a graphical calculus:

- A description of valid diagrams to be used in the calculus

- An interpretation from diagrams into some object of study

- Rules for manipulating these diagrams

Every graphical calculus we consider in this thesis will be presented in the form of a PROP[1], an idea that originated with Maclane [Mac65, §5], and neatly described as a symmetric monoidal category where the objects are the natural numbers, and the tensor product is addition. Diagrams in our graphical calculi are morphisms in such a PROP. For our diagrams this means we have a single wire-type, and boxes (representing morphisms) that take some number of wires as inputs and some number of wires as outputs. Because we are dealing with PROPs we can cross wires over, but we cannot always bend wires around: Every wire flows from an output of a box (or boundary) to an input of a box (or boundary).

Definition 2.2 (Graphical Calculus). For this thesis, a graphical calculus is a PROP with additional morphisms called *generators*. *Diagrams* are morphisms in this PROP. The calculus comes equipped with a set of *rules* which extend to an equivalence relation $\sim$ on diagrams. This PROP has an *interpretation* into some symmetric monoidal category $\mathcal{C}$, which is a strict symmetric monoidal functor $[\![\cdot]\!]$ that sends a diagram D to a morphism in $\mathcal{C}$.

Definition 2.3 (Morphisms of graphical calculi). A morphism of graphical calculi is a strict symmetric monoidal functor from one PROP of diagrams to the other. There are no requirements on the interpretations of the two calculi, and it will be made clear whether the functor respects the equivalence relation $\sim$.

The rules of a graphical calculus come in two types. There are *rewrite rules* which are expressed as 'replace this subdiagram with this other diagram' (see §2.1) and there are *meta rules*, for example 'only topology matters' in ZX. Meta rules are often used to reflect extra categorical structure that we impose on the PROP; for example the *wire-bending* rules implicit in the cups and caps of ZX are just reflections of compact closed structure on the PROP. The 'all these rewrite rules hold with the colours red and green

---

[1]A **PRO**ducts and **P**ermutations category

swapped' meta-rule from the early versions of ZX is not a rewrite rule itself but a rule that constructs rewrite rules. When we consider rules as equivalence relations we really mean the smallest equivalence relation containing all the rewrite rules *and* the meta rules. Meta rules are usually made explicit once (e.g. the (T) rule of Ref. [SZ14]) and then used implicitly from then on. Two diagrams are often referred to as being *isomorphic* if they are equivalent using only the meta rules. By contrast the rewrite rules are (ideally) explicitly mentioned every time they are used. In this thesis we indicate the rewrite rules used by the notation $A \underset{R}{=} B$, indicating that the rule R was used in transforming A into B.

Definition 2.4 (Soundness [Sel11]). A graphical calculus is sound if the interpretation $[\![\cdot]\!]$ respects the equivalence relation $\sim$. That is:

$$A \sim B \implies [\![A]\!] = [\![B]\!] \tag{2.1}$$

Definition 2.5 (Syntactic and semantic entailment). We will have cause in this thesis to consider multiple versions of a graphical calculus with the same generators (and therefore the same diagrams) but with different rules. For a graphical calculus $\mathbb{G}$ with a given set of rules $\mathcal{R}$ we may write:

$$\mathcal{R} \vdash A = B \qquad \text{or, if unambiguous, just} \qquad \mathbb{G} \vdash A = B \tag{2.2}$$

for when A is equivalent to B using the rules $\mathcal{R}$ (syntactic entailment). Likewise we may write:

$$\mathbb{G} \vDash A = B \tag{2.3}$$

if $[\![A]\!] = [\![B]\!]$ using the interpretation of $\mathbb{G}$ (semantic entailment). Note that semantic entailment is independent of any choice of rules, and syntactic entailment is independent of any choice of interpretation.

Soundness can therefore be seen as the idea that the rules (syntax) preserve the interpretation (semantics): If $\mathcal{R} \vdash A = B$ then $[\![A]\!] = [\![B]\!]$. The converse of this is *completeness*: If the semantics of two diagrams are equal, then they should also be equal from the point of view of the syntax.

Definition 2.6 (Completeness [Sel11]). A graphical calculus is complete if the equivalence relation $\sim$ reflects any equality under the interpretation. That is:

$$[\![A]\!] = [\![B]\!] \implies A \sim B \tag{2.4}$$

Combined theorems showing both soundness and completeness of a graphical calculus may be referred to as *coherence theorems* [Sel11], but we will just refer to soundness and completeness as separate properties in this work. The final property mentioned in the

introduction to this chapter is that of *universality*, the ability for a graphical calculus to represent everything it needs to. The idea is analogous to that of a universal gate set in computer science [NC10].

Definition 2.7 (Universality). A graphical calculus is called universal for a collection of morphisms if each of those morphisms can be represented by a diagram. I.e. for a collection of morphisms F:

$$\forall f \in F \; \exists D_f \text{ such that } [\![\, D_f \,]\!] = f \tag{2.5}$$

Note that just because we require a diagram to exist does not mean we have an efficient method of constructing it.

The origin of these languages was for them to be graphical calculi for categorical quantum computing [CD08]. The ZX calculus was created as a representation of manipulations of qubits (the qubits themselves represented as points in $\mathbb{C}^2$). Since then there has been some generalisation both beyond qubits (e.g. a qu**d**it version of ZX [Ran14], a qudit being a point in $\mathbb{C}^d$) and into systems over rings other than $\mathbb{C}$ (e.g. ZW_R in §2.2.3). This work remains focussed on the '-bit' case, but is not restricted to qubits:

Definition 2.8 (R-bits and qubits [Had17, Definition 5.1]). Let R be a commutative ring. The PROP of R-bit is the full monoidal subcategory of $R\,\mathbf{Mod}$ whose objects are tensor products of a finite number of copies of $R \oplus R$. Morphisms are complex matrices with 2^n columns and 2^m rows. The category of qubits is simply $\mathbb{C}$-bit and written **Qubit**.

When talking about R-bit we will sometimes use bra-ket notation (also called Dirac notation) to save space, as it allows us to clearly express the interpretations of those generators with arbitrary arity.

Definition 2.9 (Bra-kets [NC10, p62]). A ket $|\cdot\rangle$ is used to indicate a (column) vector, with a bra $\langle\cdot|$ used to indicate the dual linear functional. Tensor products of the vectors and functionals is shown by concatenation: $\langle xy| = \langle x| \otimes \langle y|$ and $|xy\rangle = |x\rangle \otimes |y\rangle$, and the composition (inner product) of the vector and functional is shown as $\langle x\,|\,y\rangle$. In our work over R-bit we will sometimes refer to the following two bases:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad\qquad |+\rangle := \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$
$$\text{and} \tag{2.6}$$
$$|1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad\qquad |-\rangle := \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

R-bit sets the scene for the calculi presented in this thesis. When considering quantum computing it is tempting to just look at qubits, but it turns out that this more general setting of R-bit still has relevance to quantum computing. For example the fragment of

Clifford+T quantum computing represents a subcategory of $\mathbb{C}$-bit. Also, when talking about the morphisms in R-bit, we will use the term *matrix* rather than *morphism* or *tensor* to indicate that we have an ordering on the inputs and outputs of each linear map, which allows us to explicitly write out the matrix (using the choice of basis implied by $R \oplus R$). Every diagram that is interpreted into R-bit therefore represents a matrix with elements from the ring R.

Remark 2.10. There is a graphical calculus called scalable ZX (SZX from here on) [CHP19] that is notable here because one can view SZX as a graphical calculus that represents (potentially enormous) ZX diagrams. That is to say that one can view SZX as a graphical calculus with diagrams and diagrammatic rules as laid out in Ref. [CHP19], with an interpretation that turns a SZX diagram into a ZX diagram. We highlight this because all the other graphical calculi considered in this thesis have interpretations that turn diagrams into matrices, as indeed SZX does in its original presentation.

2.1   Rewriting and reduction

First we shall establish notation. We will use horizontal, side-by-side placement to indicate the tensor product $\otimes$, and vertical composition (with appropriate edge connections) to indicate the usual $\circ$ composition. We shall be working **bottom-to-top** in this thesis. I.e. inputs will be at the bottom of the diagram, and outputs will be at the top. In fact there are good reasons[2] for any choice of direction, but in this thesis we are using bottom to top.

All the diagrams we will be considering in this thesis will (implicitly) be in the form of *pattern graphs* [KMS14]. Pattern graphs evolved from *string diagrams* [Pen71], via *open graphs* [DK13] (later renamed to *string graphs*). The benefit of string graphs over string diagrams is that string graphs give a combinatoric representation that is amenable to Double Pushout Rewriting (Definition 2.13). The benefit of pattern graphs over string graphs is that pattern graphs allow for !-boxes (Definition 2.23). While pattern graphs and double pushout rewriting are the backbone of graphical calculi for quantum computing, pattern graphs will not be used explicitly beyond this section of the thesis. We will instead favour the representation of !-boxes as the blue boxes found in Ref. [BK19]. We will also use ellipses to indicate repeated structure where clear. Ref. [Kis12] contains a wealth of information and results about string diagrams, string graphs and rewriting, and the following definitions are all in the style of Ref. [Kis12, §4.2].

Definition 2.11 (Rewrite Rule)**.** A graphical rewrite rule (with fixed boundary) is a pair of string graphs L and R, with a third graph I, called the *invariant subgraph* and monomorphisms $I \hookrightarrow L$ and $I \hookrightarrow R$. In this thesis we will insist that the image of I is

---

[2]Admittedly the author seems to be alone in believing that the argument for using the right-to-left direction for SZX or ZQ qualifies as a good reason

the boundary (i.e. inputs and outputs) of L and R. In particular this means that L and R have isomorphic boundaries.

Definition 2.12 (Matching). Given a rewrite rule $L \leftarrow I \rightarrow R$ we say that L *matches* onto a graph G if there is a monomorphism $m : L \hookrightarrow G$ such that the set of vertices $m(I)$ disconnects $m(L) \setminus m(I)$ from $G \setminus m(L)$. This is called the *no dangling wires* condition.

The purpose of rewriting is to cut out an instance of the graph L (found by matching L onto G) and then replace it with an instance of the graph R. We can make this idea rigorous in the following way:

Definition 2.13 (Double Pushout Rewriting [EPS73]). A Double Pushout Rewriting (DPO rewriting) of the rule $L \leftarrow I \rightarrow R$, with L matched onto G by m produces H in the below diagram. Both squares are pushouts.

$$
\begin{array}{ccccc}
L & \longleftarrow & I & \longrightarrow & R \\
\downarrow{\scriptstyle m} & & \downarrow & & \downarrow \\
G & \longleftarrow & G' & \longrightarrow & H
\end{array}
\tag{2.7}
$$

The existence and uniqueness of such a rewrite is covered in Ref. [KMS14], based on the results of Ref. [DK13].

Now that we can rewrite graphs we shall touch on how to perform *reductions*. This idea originates in that of *term rewriting*, see Ref. [BN98]. Term rewriting combines logic and algebra to give a system of applying modifications to an algebraic term (e.g. $X^2 + 2 + 2X^2 - 1$) in order to find an equivalent term with certain properties (e.g. $3X^2 + 1$, which has grouped similar monomials together). Term rewriting is an essential part of conjecture synthesis, but this work does not make use of any notable changes to the term reduction system already present in earlier work [Kis13; Kis12].

Definition 2.14 (Graphical reduction and reducibility [Kis12]). Given an ordering ω on the diagrams of a graphical calculus we say a rule application $A \mapsto B$ is a reduction if $\omega(A) > \omega(B)$. A diagram A is called reducible (or a *redex*) by a set of rules $\mathcal{R}$ if a single application of one of the rules in $\mathcal{R}$ yields a reduction of A.

2.2   A zoo of calculi

We will now define the generators and interpretations of the families of graphical calculi we will use in this thesis. We will not provide all the rules for each calculus here, just the ones we will have reason to reference on occasion. This is because several different rulesets exist for each calculus, but also because any time we will need a particular ruleset we will state the rules explicitly. Note that the calculi ZQ and RING$_R$ (introduced in this thesis) are not covered in this background chapter, and are instead given their own chapters (§4 and §6).

2.2.1   Wires

Definition 2.15 (Wires, cups, and caps [CK17, §4, §5])**.** (with thanks to Ref. [Sel07] for the standardisation of much of the language and notation). All of the calculi considered here are based on the same underlying structure of wires. These wires represent the structure of a compact closed PROP over R-bit without any additional morphisms. There will be a section in this thesis that does not assume compact closure, but we have sign-posted this clearly (see Definition 5.1). In ZX, ZW and ZH (and, when we get to them, RING and ZQ) we have the following graphical elements:

$$\text{wire} \qquad \Big| \quad \overset{\llbracket \cdot \rrbracket}{\longmapsto} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.8}$$

$$\text{swap} \qquad \asymp \quad \overset{\llbracket \cdot \rrbracket}{\longmapsto} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.9}$$

$$\text{cap} \qquad \frown \quad \overset{\llbracket \cdot \rrbracket}{\longmapsto} \quad \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \tag{2.10}$$

$$\text{cup} \qquad \smile \quad \overset{\llbracket \cdot \rrbracket}{\longmapsto} \quad \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}^T \tag{2.11}$$

$$\text{empty} \qquad \boxed{} \quad \overset{\llbracket \cdot \rrbracket}{\longmapsto} \quad \begin{pmatrix} 1 \end{pmatrix} \tag{2.12}$$

Note that the swap, in bra-ket notation, has the action of $|\,xy\,\rangle \mapsto |\,yx\,\rangle$, and that our cups and caps obey the *snake equations*:

$$\overset{\text{}}{} = \Big| = \overset{\text{}}{} \tag{2.13}$$

For the rest of this section we won't mention the wires when defining the calculi.

2.2.2   ZX

Officially ZX is called 'the ZX-calculus' but since there is no risk of ambiguity, and it is common to hear people do so, we shall just use the short form (likewise 'ZW' and 'ZH'). We shall, however, indicate *fragments* of each calculus using either words or subscripts. For example 'Clifford+T ZX', or '$ZX_{\pi/4}$'. ZX originated in Ref. [CD08], and since then has flourished into the most recognisable of the calculi presented here. We covered part of its story in §1, and given the range of definitions and fragments there is still debate as to whether there is 'a' or 'the' ZX calculus, or even what requirements a graphical calculus would need to fulfil to count as a ZX calculus[3].

Definition 2.16 (ZX)**.** The ZX-calculus comes in several fragments. Each fragment is determined by the phases (labels) allowed on the nodes. In all cases the phases form a group that is a subgroup of $[0, 2\pi)$ under addition. Certain special subgroups have special names (given below) and only certain subgroups have been the subject of study.

Group	Name	Shorthand	Completeness reference
$[0, 2\pi)$	Universal	ZX_U	[NW17]
$< \pi/2 >$	Stabilizer	$ZX_{\pi/2}$	[Bac16]
$< \pi/4 >$	Clifford+T	$ZX_{\pi/4}$	[JPV18a]
G	-	ZX_G	[JPV19]

Note that Ref. [Bac16] combines the scalar and scalar-free completeness proofs by the same author. The interpretation of a ZX diagram is understood as a complex matrix, even though only the Universal fragment of ZX is universal onto all complex matrices. When a generic group G is considered it is always assumed to contain $\pi/2$, and usually assumed to contain $\pi/4$. We now present the generators, interpretation and sample rules for ZX:

- Green (Z) and red (X) spiders with any number of inputs and outputs (including 0):

$$\overset{[\![\,\cdot\,]\!]}{\longmapsto} |0\ldots0\rangle\langle0\ldots0| + e^{i\alpha}|1\ldots1\rangle\langle1\ldots1| \tag{2.14}$$

$$\overset{[\![\,\cdot\,]\!]}{\longmapsto} |+\cdots+\rangle\langle+\cdots+| + e^{i\alpha}|-\cdots-\rangle\langle-\cdots-| \tag{2.15}$$

Phases of 0 are instead left blank.

---

[3]Private communication with Niel de Beaudrap and Harny Wang

- Hadamard gates [NC10, p19] have one input and one output:

$$\left\llbracket\cdot\right\rrbracket \longmapsto \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.16}$$

There is a trend, welcomed by the author for its clarity, to draw edges decorated with Hadamard gates as dashed, blue lines (in situations where there is neither ambiguity nor loss of rigour).

$$\vdots := \quad \tag{2.17}$$

Hadamard gates are derivable from red and green spiders [DP09] but, as we shall see below, red spiders can be constructed from green spiders and Hadamard gates. It has become a matter of authorial preference whether a given paper will consider ZX to be constructed from red and green spiders, or from green spiders and Hadamard gates (in a manner similar to that of graph states), or even all three.

- Meta-rules:

  - Only topology matters (the underlying structure is an open graph)
  - All rules hold with red and green swapped

- Example rewrite rules. We present them here using !-boxes, as is standard, and a formal definition is given later as Definition 2.23. These light blue boxes indicate '0 or more copies of the subdiagram', requiring the same number of copies for the matching box on the other side of the equation. For example if the !-box surrounding the input wire on the left hand side of the rule Spider 1 is expanded as having 3 copies, then the box around the inputs on the right hand side of Spider 1 is also understood to indicate 3 copies.

$$\tag{2.18}$$

$$\tag{2.19}$$

$$
\begin{array}{cc}
\underset{\text{Colour swap}}{=} & \text{(2.20)}
\end{array}
$$
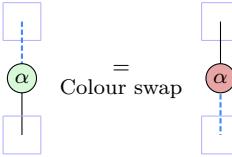
2.2.3   ZW

The ZW calculus [Had15; Had17] was created for manipulating the GHZ and W entanglement states, representing the two classes of connected entanglement available for three qubits. The W state (generalised to any arity) is represented as a black spider. The GHZ state, too, is generalised in terms of arity, but also given a phase, becoming a Z spider similar to the Z spider of ZX. We will introduce a convention of drawing nodes that never contain a phase (e.g. the black, rounded spider of ZW) smaller than nodes that can contain a phase (e.g. the white, rounded spider of ZW).

In contrast with ZX, which aims to represent only qubit quantum computing, ZW has more freedom. By allowing the phases of the Z spider to come from some choice of ring R, the diagrams in ZW_R (as that fragment is then called) are universal over R-bit. Not only that, but the rules of ZW_R remain sound and complete, independent of the choice of R. To recover qubit quantum computing one simply sets R to $\mathbb{C}$, but it should be noted that completeness of the Clifford+T [JPV18a] and Universal [NW17] fragments of ZX were shown by equivalences with two different fragments of ZW. ZW's own completeness comes from the existence of a normal form [Had15].

Definition 2.17 (ZW [Had17])**.** The ZW calculus is generated by Z (white) and W (black) spiders. It is parameterised by a choice of commutative ring R, with an interpretation into R-bit.

- The Z (white) spider, parameterised by $r \in R$, and the W (black) spider:

$$
\xrightarrow{\llbracket \cdot \rrbracket} \; |\,0\ldots0\,\rangle\langle\,0\ldots0\,| + r\,|\,1\ldots1\,\rangle\langle\,1\ldots1\,| \tag{2.21}
$$
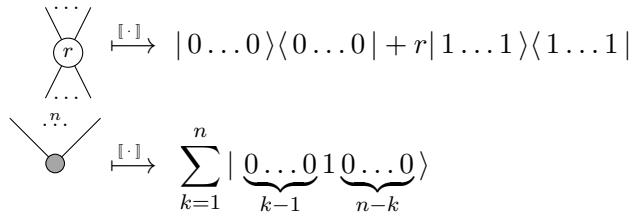
$$
\xrightarrow{\llbracket \cdot \rrbracket} \; \sum_{k=1}^{n} |\, \underbrace{0\ldots0}_{k-1}\,1\,\underbrace{0\ldots0}_{n-k}\,\rangle \tag{2.22}
$$

When the phase on the white node is 1 it is left blank.

- The Crossing x is a non-commutative, derived generator of fixed arity used in the rules:

$$\vcenter{\hbox{}} \xmapsto{\;[\![\cdot]\!]\;} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.23}$$

- Meta-rules:

    - Only topology matters (the underlying structure is an open graph)

- Example rewrite rules:

$$(2.24)$$

$$(2.25)$$

$$(2.26)$$

2.2.4   ZH

The 'Z' of ZW, ZX, and ZH (and indeed ZQ) are all minor modifications on the same theme. The 'H' of ZH refers to the H-box; a generalisation of the Hadamard gate in both arity and phase. One important notational aspect is that the Hadamard node of ZX (a yellow rectangle with no phase) is represented in ZH as a white rectangle with a phase of -1, which can feel counter-intuitive when moving between the calculi. Similar to ZW, the completeness of ZH was proven via the existence of a normal form for its diagrams.

Definition 2.18 (ZH [BK19])**.** The ZH calculus is generated by phase free Z spiders and generalised (in both arity and phase) Hadamard nodes called 'H-boxes' [BK19]. The original paper only allows the H-boxes to have phases from the complex numbers, but one of the results in this thesis (§4.4.1) is that this can be extended to any commutative ring containing the element $\frac{1}{2}$, preserving soundness and completeness. We will give the version of ZH from Ref. [BK19] below, our only modification being that we refer to it as $\text{ZH}_{\mathbb{C}}$:

- The Z (white) spider and H-box (rectangle with m inputs and n outputs) generate ZH$_{\mathbb{C}}$:

$$\xmapsto{\;[\cdot]\;} \;|\,0\ldots0\,\rangle\langle\,0\ldots0\,| + e^{i\alpha}|\,1\ldots1\,\rangle\langle\,1\ldots1\,| \tag{2.27}$$

$$\xmapsto{\;[\cdot]\;} \;\sum_{\text{bitstrings}} c^{i_1\ldots i_m j_1\ldots j_n}|\,j_1\ldots j_n\,\rangle\langle\,i_1\ldots i_m\,| \tag{2.28}$$

The sum in the interpretation of the H-box is over all bitstrings $i_1\ldots i_m j_1\ldots j_n$. The reader may find it easier to view this interpretation as a matrix where every entry is 1, with the exception of the bottom right entry which is c. In the case where this matrix only has one row and one column (i.e. a scalar) the sole entry is $1 + c$. When the phase is -1 it is left off the H-box.

- The derived generators of ZH include the grey spider and the NOT gate:

$$\tag{2.29}$$

$$\tag{2.30}$$

- Meta-rules:

    - Only topology matters (the underlying structure is an open graph)

- Example rewrite rules:

$$\tag{2.31}$$

$$\tag{2.32}$$

2.3   Decorated and simple diagrams

The presentation of the calculi above (§2.2) already involved using variables to indicate arbitrary phases. The meaning of this is usually considered to be so clear that it is not worth remarking upon: The calculus ZH$_{\mathbb{C}}$ was introduced in Ref. [BK19] using variables

on the phases. We will, however, have cause to be rather pedantic about this nuance in this thesis. Our reason for caring about this nuance is that when we later try to generate and generalise theorems it is important to be able to introduce and verify these useful components (see §3). First we will clearly define phase variables and !-boxes, and then declare a diagram to be *simple* if it does not use them.

Definition 2.19 (Phase Algebra). The phase algebra for a vertex in a graphical calculus is the collection of terms allowed to be written as a phase for that vertex.

Example 2.20 (ZX Phase Algbera). ZX uses a phase *group* algebra, which reflects the fragment of the calculus (Definition 2.16). For example, Universal ZX has phase group $[0, 2\pi)$ under addition, and the red and green spiders can be decorated with terms from that algebra. The following are therefore valid Universal ZX diagrams:
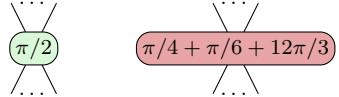
$$\begin{array}{cc} \boxed{\pi/2} & \boxed{\pi/4 + \pi/6 + 12\pi/3} \end{array} \tag{2.33}$$

The following equation is not a Universal ZX diagram, because the group algebra does not contain the symbol '$\times$':
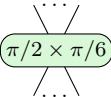
$$\boxed{\pi/2 \times \pi/6} \tag{2.34}$$

Definition 2.21 (Phase variables). A phase variable is a formal variable introduced to the phase algebra. For ZX (i.e. phase group variables) we tend to use $\alpha, \beta, \ldots$, and for ZH and ZW (i.e. phase ring variables) use $a, b, c, \ldots$ or $r, s, \ldots$. We can *evaluate* a phase variable to some element of the phase algebra by replacing all instances of that phase variable in the diagram with that element.

Example 2.22 (ZX Phase Variable). The following are considered to be Clifford+T ZX diagrams but with the phase variable α:
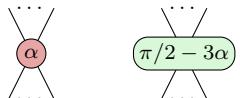
$$\begin{array}{cc} \boxed{\alpha} & \boxed{\pi/2 - 3\alpha} \end{array} \tag{2.35}$$

Note that this is indistinguishable from a ZX diagram where the phase group is generated by $\pi/4$ and α. We will use this observation when talking about parameter spaces in §7.

Now that we have covered phase variables let us look at !-boxes. !-boxes indicate repeated elements in a diagram, and are notational sugar[4] for the concept of pattern graphs introduced in Ref. [KMS14].

---

[4]The region indicated by a !-box is less visually cluttered than drawing out the !-vertices and directed edges of a pattern graph.

Definition 2.23 (!-box [KMS14]). A !-box (pronounced 'bang box') indicates a collection of nodes (potentially including wire vertices) in a diagram. By an instantiation of a !-box we mean choosing a number $n \in \mathbb{N}$ and making n copies of those nodes, preserving their connectivity. A !-box is drawn as a blue box surrounding the nodes inside the !-box.

Example 2.24 (!-boxes in ZH [BK19, §2.3]). The diagram below represents an entire family of ZH diagrams:

$$\tag{2.36}$$

Definition 2.25 (Simple and decorated). A diagram that does not contain any phase variables or !-boxes is called simple.

Any diagram that isn't simple represents a family of simple diagrams, and it is important to be able to recover any particular simple diagram from this family. When talking about these families we will use the term 'parameterised family', and be explicit, where possible, about what these parameters are. This isn't just to say 'parameterised by !-boxes and phase variables', but to give each of these a name; phase variables are simply referred to by their own name (e.g. the phase variable α is still referred to as α) and for !-boxes we just ascribe to each !-box the name $\delta_1, \delta_2, \dots$. With explicit names for the parameters we can recover a simple diagram by *instantiating* each !-box some number of times, and *evaluating* each phase variable to some element of the phase algebra. We will use the notation $\alpha | \alpha = \pi$ to indicate that α is a parameter, and that it should be evaluated to the value π. When we are talking about equations it is understood that the evaluation and instantiation is to be applied to both the left and right hand side of the equation.

Example 2.26 (An explicit evaluation and instantiation). The (slightly simplified) spider law in Universal ZX is parameterised over:

- !-boxes: $\delta_1 \in \mathbb{N}$ inputs and $\delta_2 \in \mathbb{N}$ outputs

- phase variables: $\alpha_1, \alpha_2 \in [0, 2\pi)$

We write this parameterised family of equations as:

$$\tag{2.37}$$

We now instantiate some of the parameters of the spider law, resulting in what is still an infinite, parameterised family:

$$
\left\{
\begin{array}{ccc}
\boxed{}\ {}^{\delta_1} & & \boxed{}\ {}^{\delta_1} \\
\bigl(\alpha_1\bigr) & = & \bigl(\alpha_1 + \alpha_2\bigr) \\
\bigl(\alpha_2\bigr) & & \\
\boxed{}\ {}^{\delta_2} & & \boxed{}\ {}^{\delta_2}
\end{array}
\right\}_{\alpha_1,\alpha_2,\delta_1,\delta_2 | \alpha_1 = \pi, \delta_1 = 2}
\tag{2.38}
$$

$$
= \left\{
\begin{array}{ccc}
\boxed{}\ {}^{\delta_1} & & \boxed{}\ {}^{\delta_1} \\
\bigl(\pi\bigr) & = & \bigl(\pi + \alpha_2\bigr) \\
\bigl(\alpha_2\bigr) & & \\
\boxed{}\ {}^{\delta_2} & & \boxed{}\ {}^{\delta_2}
\end{array}
\right\}_{\alpha_2,\delta_1,\delta_2 | \delta_1 = 2}
\tag{2.39}
$$

$$
= \left\{
\begin{array}{ccc}
\bigl(\pi\bigr) & = & \bigl(\pi + \alpha_2\bigr) \\
\bigl(\alpha_2\bigr) & & \\
\boxed{}\ {}^{\delta_2} & & \boxed{}\ {}^{\delta_2}
\end{array}
\right\}_{\alpha_2,\delta_2}
\tag{2.40}
$$

When the topic of verification comes up later (§8) the goal is to be able to say that some family of equations holds true, just by checking whether some subset of that family holds true.

Definition 2.27 (Soundness for an equation)**.** An equation of diagrams $\mathbb{D}_1 = \mathbb{D}_2$ is sound if $[\![\, \mathbb{D}_1 \,]\!] = [\![\, \mathbb{D}_2 \,]\!]$. A family of equations is sound if every equation in the family is sound.

Definition 2.28 (Verifying family)**.** A family of equations S is said to verify the family of equations S' if S being sound implies that S' is sound:

$$
e \text{ sound } \forall e \in S \quad \implies \quad e' \text{ sound } \forall e' \in S'
\tag{2.41}
$$

One of the questions considered later in this thesis is 'when can a parameterised family of equations be verified by a small set of simple equations?' Our reasons for wanting to answer this will hopefully be made clear in the next chapter, where we give the background for conjecture synthesis.

Chapter 3

Conjecture Synthesis

In this chapter:

▷ We discuss conjecture synthesis for terms and string graphs

▷ We introduce a novel step in the conjecture synthesis process

▷ We discuss the difficulties associated with our graphical calculi

▷ We give an example from a conjecture synthesis run

It was mentioned in the introduction that IsaCoSy (the Isabelle Conjecture Synthesis) was created with the '[...] aim to automatically produce a useful set of theorems'. The full quotation, however, is:

> Given a set of initial definitions of recursive datatypes and functions, we aim to automatically produce a useful set of theorems, that will be useful as lemmas in further proofs, by either a human or an automated theorem prover. (IsaCoSy: Synthesis of inductive theorems [JDB09])

Where this thesis deviates from that work is in the data structures considered (for us it will be diagrams rather than recursive datatypes) and in the introduction of a *generalisation* step (to be discussed later). Where that work deviated from its predecessors was in terms of method. IsaCoSy follows a *generative* approach, where hypotheses are generated and then assessed, compared to a *deductive* approach where new theorems are deduced from old (see Figure 3.1 for a diagram indicating the deductive approach).

The generative approach relies on the concept of reduction, a concept that was briefly covered in §2.1. For IsaCoSy this meant term reduction, but for this work it will mean string graph reduction. The key in the generative approach is to not generate (or to generate and immediately discard) any *reducible* objects, be they terms or string graphs.
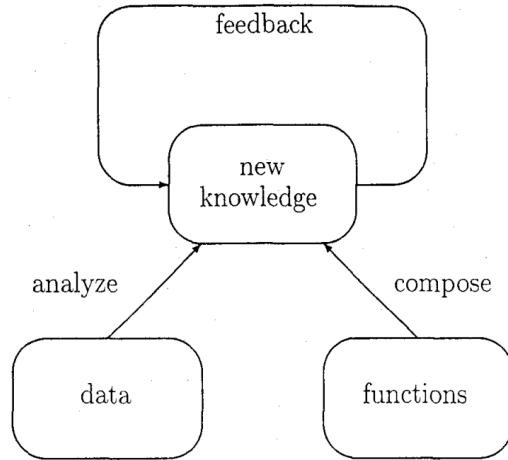
Figure 3.1: A diagram captioned 'The construction of new knowledge' from Ref. [Amm93], indicating the deductive approach to automated theorem generation

Example 3.1 (Adding 0 is pointless [JDB09, Example 1]). The following is a term reduction describing part of the nature of $+$ acting on the natural numbers:

$$0 + x \longrightarrow x \tag{3.1}$$

We therefore know that if we generate an expression containing $0 + x$ it would have been both simpler and equivalent to generate the same expression just containing x instead.

While Example 3.1 feels entirely obvious this simple idea works for any abstract reduction system. What's more, conjecture synthesis can generate new reductions: Every orientable theorem results in a new reduction that will then further limit the space that needs to be searched. In contrast to the deductive model of Figure 3.1 we can therefore use the generative model of Figure 3.2. This model has been implemented in the QuantoCoSy part of Quantomatic [KZ15]. The test for how well the model works is (counter-intuitively) to see how many *fewer* theorems it creates for a given system. Any theorems it doesn't provide to the user as output were theorems that were reducible. Earlier work with an earlier version of QuantoCoSy [Kis13] claimed that it produced exponentially fewer rules when considering GHZ/W (later to become ZW) diagrams up to a certain size:

> The results there were promising, as we demonstrated an exponential drop-off in the number of extraneous rules generated when using [a reducible object]-eliminating routine as compared to a naive synthesis routine (Synthesising Graphical Theories [Kis13])

The starting point for this research was clear: Use QuantoCoSy to generate theorems in ZX or ZW. The big question remaining was 'how should we measure success?' For this
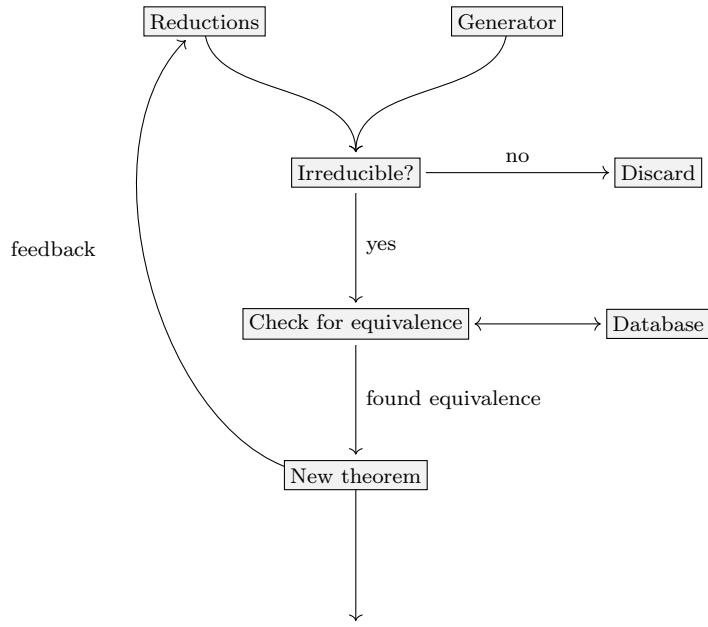
Figure 3.2: The generative model, as used in QuantoCoSy. The database stores the matrix representation of every irreducible diagram generated. This is then used for comparison with each newly generated diagram to determine whether a new equivalence has been found.

the author took the goal of designing a system that could 'feasibly discover a complete ruleset for ZX or ZW, given elementary domain knowledge'. This goal was chosen before the complete rulesets for Clifford+T or Universal ZX were discovered, but this goal, together with the question of implementation, led to the research in this thesis.

One final note is that in our conjecture synthesis runs we usually considered two diagrams to be equivalent if they agreed up to non-zero scalar. This is because scalars are comparatively easy to keep track of using a single complex number rather than a diagram.

3.1   Generating diagrams

As research for this thesis started, the ZX-calculus (see §2.2.2) was already gaining attention. The book Ref. [CK17] was finished, focussing almost exclusively on ZX, and the quantum computing lectures at the University of Oxford (attended by the author) were lectured using ZX. ZX had already been shown to be complete for stabilizer quantum mechanics, although not for the other fragments, and the expectation and hope was that it would be shown to be complete for Clifford+T and universal quantum computation soon (see Table 1.3). ZX, for our purposes, is generated by red and green spiders, and these spiders are very relevant for CoSy. We shall take the following as our initial 'elementary domain knowledge':

- Spiders have arbitrary arity

- Adjacent spiders of the same colour fuse

- If two edges join a pair of spiders, then those spiders either fuse by the spider law, or the edges disconnect by the Hopf law

Combining these three points results in the realisation that ZX diagrams can be reduced by the spider and Hopf laws to a labelled, bipartite graph. Since any reducible diagram will be ignored by the model in Figure 3.2 we can instead simply generate labelled bipartite graphs (the label just contains the information on the colour and angle of the spider). For this QuantoCoSy uses the algorithm given by Ref. [CR79]. ZW also contains two colours of spider, but crucially we cannot use the spider laws of ZW (Equation (2.24)) to reduce all diagrams to bipartite graphs. Likewise when the calculus ZH (Definition 2.18) was introduced it too had spider laws, similar to ZW's, that do not provide us with a reduction to bipartite graphs. For this reason the first implementations focused on ZX.

Remark 3.2. Thanks to the combinatoric model of pattern graphs it is possible to generate diagrams containing !-boxes. One of the contributions of this thesis is that we can now verify equations involving !-boxes without needing to directly reason with !-boxes, see §8.4. This verification depends on a property called 'separability', covered in §8.4.1, which is a condition one could build into the pattern graph generator.

3.2   Generalising theorems

Figure 3.3 contains one of the first ZX rules generated by QuantoCoSy running on Duvel.[1] Note that at this point the feedback loop from Figure 3.2 had not been implemented, and scalars were still being considered. This theorem was, however, towards the upper end in terms of the size of diagram considered for those runs.[2]
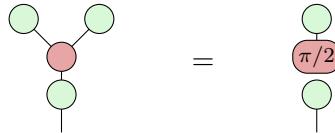
Figure 3.3: An early example of a theorem generated by Duvel. In this case the theorem is '-1001866680_-1614245455.qrule' from run '18-04-04-10-55-19-duvel', chosen because it is first in the list when displayed by Quantomatic.

---

[1]The computing cluster made available for this research, with 24 cores at 2.9MHz.

[2]The results of these QuantoCoSy runs can be found as additional files provided along with this thesis, although we do not rely upon them for any of the results in this thesis. Results are in folders labelled by a date and target hardware, and there is a file 'summary.txt' in each folder recording the notes for that run.

Remark 3.3 (The generalisation step). The first contribution of this thesis is that of adding a *generalisation step* to the method of generative conjecture synthesis. It is performed during the feedback step of Figure 3.2, and during this step we take our newly synthesised result and try to generalise it as much as possible before creating new reductions from it. This inner generalisation algorithm could itself be deductive (we cover examples in §7.1) or produce informed guesses which are then verified (see §7.2.4). In both cases the processes are novel to this thesis, and the non-deductive process is only possible due to a novel link between phase algebras and algebraic geometry.

Example 3.4 (Spotting a generalisation). the theorems in Figure 3.4 form part of a wider pattern, illustrated by the use of !-boxes in Figure 3.5. Likewise the two 'π-commutations' exhibited in Figure 3.6 are both instances of the more general rule shown in Figure 3.7.

This question of how to generalise theorems is not new. The philosophical 'Problem of Induction' of creating ideas from evidence dates back to Hume in 1739 [Hum03]. Example 3.5 gives a more recent example from the field of conjecture synthesis, where the task of term generalisation is handled by attempting to replace existential quantifiers with universal quantifiers, and if that proves to be too general then successively greater restrictions are placed on the universal quantifier. For graphical calculi we have two ways to generalise. We can generalise these graph patterns with !-boxes (as in Figure 3.5) and generalise the phase patterns with 'phase variables' (as in Figure 3.7).

Example 3.5 (Term generalisation using quantifiers [Amm93, p416]). The progression of first attempting to generalise the variable n to universal quantification, and then to restricted universal quantification in a conjecture about natural numbers and primes.

$$\exists n, p, q(n \in N \land p \in P \land q \in P \land n = p + q) \tag{3.2}$$

$$\forall n(n \in N \to \exists p, q(p \in P \land q \in P \land n = p + q)) \tag{3.3}$$

$$\forall (n(n \in N \land \text{is-even}(n) \to \exists p, q(p \in P \land q \in P \land n = p + q)) \tag{3.4}$$

The question of how to introduce phase variables (and, to a lesser degree, !-boxes) is covered in §7, and relies on a novel link between phase algebras and algebraic geometry. We will discuss how to then verify these conjectures in §8. One of the interesting results of combining these two ideas arises in §8.3, where we note that some phases can, almost automatically, be replaced with more general phase variables. In terms of Example 3.5 this would amount to saying that certain existential quantifiers could automatically be replaced with universal quantifiers. The operational mantra for this is that there is a bound on the algebraic complexity for the phases in an equation; if a phase appears less often than the degree of its minimal polynomial then it can be replaced by a phase variable (Remark 8.36).

The framework of §7 is, however, limited by the choice of phase algebra. For ZX the framework can infer all the rules in the complete ruleset of Stabilizer ZX from Ref. [Bac14a] and Clifford+T ZX in Ref. [JPV18a], but it cannot reason about the rules needed for completeness of Universal ZX: The original completeness result contained the

(AD) rule [NW17] which requires the user to compute the magnitude and argument of $\lambda_1 e^{i\beta} + \lambda_2 e^{i\alpha}$. With considerable work that ruleset was reduced to that of Ref. [Vil19], which requires the user to compute the following:

> In rule (EU'), β_1, β_2, β_3 and γ can be determined as follows: $x^+ := \frac{\alpha_1 + \alpha_2}{2}$, $x^- := x^- \alpha_2$, $z := -\sin(x^+) + i\cos(x^-)$ and $z' := \cos(x^+) - i\sin(x^-)$, then $\beta_1 = \arg z + \arg z'$, $\beta_2 = 2\arg(i + \frac{|z|}{|z'|})$, $\beta_3 = \arg z - \arg z'$, $\gamma = x^+ - \arg(z) + \frac{\pi - \beta_2}{2}$ where by convention $\arg(0) := 0$ and $z' = 0 \implies \beta_2 = 0$. (Figure 2, A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics [Vil19])

Since we are always striving to find more refined, or even just different, complete rulesets for these calculi it is possible that there will exist some complete ruleset for ZX where every rule *does* fit into the framework presented in §7. The authors of Ref. [JPV18b], however, are pessimistic about this in their comment: 'Notice however that [SZ14] and [Jea+17] are two different kinds of evidence that such a finite complete axiomatisation [using linear rules] may not exist'. (See Remark 7.20 for how the linear rules of Ref. [JPV18b] coincide with the framework for ZX described in §7.)

Constructing a framework that allows the direct inference of the (EU') rule's external computations, also called *side conditions*, seems infeasible to the author. This in no small part prompted the wider research of this thesis. Fortunately another avenue opened up, for despite the apparent complexity of the (EU') rule of Ref. [Vil19] it actually hints at something far simpler: It is analogous to the Euler angle decomposition of rotations in 3D space (this is the origin of the name (EU)). In response to this the author constructed a new, complete, graphical calculus called ZQ (Definition 6.14) which took all rotations to be primitive, and not just the Z and X rotations of ZX. In contrast to Universal ZX it *is* possible to express a complete ruleset of (the equally universal and complete calculus) ZQ using the linear framework of §7.

Figure 3.4: Two examples of (scalar-free) copying
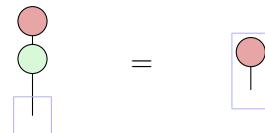
Figure 3.5: The more general version of (scalar-free) copying, using !-boxes
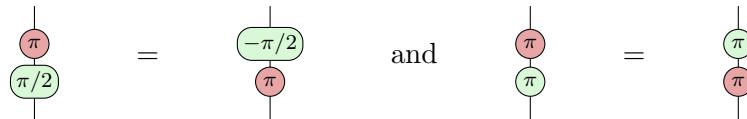
Figure 3.6: Two examples of (scalar-free) π-commutation

Figure 3.7: The more general version of (scalar-free) π-commutation

3.3   Generalising ring calculi

By this point in the narrative the paper introducing ZH (Ref. [BK19]) had been published. The author was then presented with the question of which of ZH and ZW, both being ring-based, spider calculi, would be better to investigate. Both ZH and ZW are built from two types of spider, with the rules of ZH involving a third, derived, type of spider. Neither calculus relies on side conditions for its complete ruleset. ZW, unlike ZH, had the advantage of having been used directly for the completeness results for Clifford+T and Universal ZX, but the completeness results of both ZH and ZW were shown via normal forms.

It struck the author that the similarities between ZH and ZW were more interesting than their differences, and so at this point the research diverted from synthesising conjectures in these ring-based languages to trying to find a unifying system for discussing them. From this came the RING$_R$ graphical calculus (§4.1); a calculus designed to be as generic a phase ring calculus as possible. By construction it is easy to translate RING$_R$ diagrams into ZW or ZH diagrams, but by considering the category of phase ring graphical calculi we can actually say that RING$_R$ is essentially initial for graphical calculi with phase ring R (Corollary 5.14). RING$_\mathbb{C}$ has even been described as 'solving the field' for phase ring qubit graphical calculi[3].

The investigation of RING$_R$ and its categorical context (§5) gave rise to the idea of phase homomorphisms and phase homomorphism pairs; structures that acted directly on the phase of the diagrams while also commuting with the interpretations. These phase homomorphism pairs also arise as symmetries in the framework introduced in §7, and are instrumental in the finite verification result of §8.3. At the end of §5 we look at the corresponding structures for phase group calculi, and then generalise further to Σ-calculi.

This ends our introductory chapters, and so we shall begin by introducing the calculus RING.

---

[3]Aleks Kissinger, private communication

Chapter 4

The Graphical Calculus RING

In this chapter:

> ▷ We introduce a complete, universal phase ring calculus, $\textsc{ring}_R$

> ▷ We extend $\text{ZH}_\mathbb{C}$ to the more general ZH_R

> ▷ We exhibit translations between ZW_R, ZH_R, and $\textsc{ring}_R$

> ▷ The calculus $\textsc{ring}_R$ introduced in this chapter is both a motivation and an example used throughout the rest of this thesis

The results of this chapter were born from the idea of a generic phase ring calculus for conjecture synthesis. The ZW calculus 'contains rational arithmetic' in the words of Ref. [Coe+11], meaning that one can use the W and GHZ states of ZW to perform addition and multiplication. This is not to say that ZW is the only way of encoding a ring-like structure in a graphical calculus, or even that there are not other ways of presenting ring-like structures inside ZW. Just because ZW contains rational arithmetic does not mean in embodies rational arithmetic.

ZW is not the only calculus with a phase ring. We were introduced to $\text{ZH}_\mathbb{C}$ in Definition 2.18, which is universal and complete for **Qubit** [BK19], but there is also the newly introduced Algebraic ZX[1] [Wan19], which coerces the phase group of ZX into a phase ring.

Our aim for this chapter is to introduce a phase ring calculus that lays bare the ring structure of the phase algebra. This will allow us to explore the uses and limitations of such a phase algebra, especially over qubits, in later chapters.

We introduce our own calculus $\textsc{ring}_R$ in §4.1, which is designed with the ring operations first and foremost, and to work for any commutative ring R. We exhibit a complete ruleset for $\textsc{ring}_R$ in §4.2, found via a translation with ZW_R. We then extend ZH to work for arbitrary commutative rings with $\frac{1}{2}$ in §4.4.1, and also a translation between $\textsc{ring}_R$

---

[1]The author would have suggested a different name, such as 'Ring ZX'.

and this more general version of ZH. Although the translation with ZH_R is covered in far briefer terms than the translation with ZW_R we use it to produce a different, complete, ruleset for RING_R in §4.4.3 (provided the ring R contains $\frac{1}{2}$).

As mentioned earlier the calculus RING is used as both motivation and example throughout the rest of this thesis, but the only knowledge required for those future chapters is familiarity with the definition and main results of §4.1, and awareness of the rules in Figure 4.3.

4.1   The definition of RING

The graphical calculus RING_R is created to be 'as generic a phase ring graphical calculus over states as possible'. With this in mind our generators are precisely the ring operations of some ring R, a set of states representing the elements of R, and the ambient wires, swaps, cups and caps of a compact closed graphical calculus. There are complete rulesets available via translation from both ZW_R (§4.2) and also the extended ZH calculus, ZH_R, which we introduce in §4.4.

Definition 4.1 (RING_R). The graphical calculus RING_R for a commutative ring R is a (compact closed) PROP with generators and interpretation given by Figure 4.1, with derived generators given in Figure 4.2, and rules given in Figures 4.3 and 4.4.

Let's unpack this definition a little. The generators in Figure 4.1 form the core of the calculus. Every diagram in RING is built from these three components. Since we allow wire bending (i.e. this is a compact closed calculus) these generators can be plugged together with no regard for concepts of 'input' or 'output'. The multiplication gate is the familiar (phase-free) Z spider from ZX, ZW, and ZH, and so retains all of its spider-like qualities (commutativity, associativity, etc.). Where RING differs from some of the other calculi is that the addition gate has a distinguished wire, drawn at the top in Figure 4.1. This means that addition in RING forms a 'one-sided' spider: wires connected to the rounded side of the gate act like wires connected to a spider, but the distinguished wire (attached to the flat, bold side) does not.

Readers familiar with monoids should note that the multiplication gate forms a monoid with the state 1 as a unit, and the addition gate forms a monoid with the state 0 as a unit. Knowing this allows us to create the derived generators of Figure 4.2. Derived generators are built from our core generators, and represent motifs that appear often enough in our diagrams to warrant creating these shorthand notations. The Hadamard Gate in ZX is a derived generator: It is derivable from Z and X spiders, and so is technically redundant, but is so common and useful a gate that it is afforded the same status as a generator. In RING the multiplication gate can be generalised as the familiar, phased Z spider (as in ZW or ZX), and the addition gate as a phased, one-side spider. In either case the label on the derived generator is an element of the ring S. To summarise:

Remark 4.2 (Topology matters a bit). The white multiplication node is indeed a spider, in the sense of Ref. [CD08], but addition has a distinguished output indicated by the bold, flat edge.

The two other derived generators in Figure 4.2 are the NOT gate and Hadamard gates respectively. These derived generators use the 'flipped' version of the addition gate, drawn upside down. This is simply a space-saving convention: The distinguished wire comes from below, and so the bold, flat edge of the gate is drawn at the bottom. To calculate the interpretation we would use bent wires to allow us to draw the addition gate the same way up as it is in Figure 4.1.

Now that we have defined our language we shall explore its important properties: Completeness, Soundness and Universality.

Theorem 4.3 (RING is complete)**.** *The language* RING_R *is complete for any commutative ring R.*

Proof. We include in Figures 4.5 and 4.6 a translation between the generators of RING_R and ZW_R that preserves the interpretation. In §4.3 we derive, in RING_R, each equality $F_{\mathrm{RING}}L = F_{\mathrm{RING}}R$ for each rule $L = R$ in [Had17; Had15]. In Lemma 4.37 we derive the equation $g = F_{\mathrm{RING}}F_{\mathrm{ZW}}(g)$ for each generator g of RING_R. Therefore for each sound diagrammatic equation $D_1 = D_2$ we will be able to derive:

$$\mathrm{ZW}_R \vdash F_{\mathrm{ZW}}D_1 = F_{\mathrm{ZW}}D_2 \tag{4.1}$$

$$\mathrm{RING}_R \vdash F_{\mathrm{RING}}F_{\mathrm{ZW}}D_1 = F_{\mathrm{RING}}F_{\mathrm{ZW}}D_2 \tag{4.2}$$

$$\mathrm{RING}_R \vdash D_1 = F_{\mathrm{RING}}F_{\mathrm{ZW}}D_1 \qquad \text{and} \qquad D_2 = F_{\mathrm{RING}}F_{\mathrm{ZW}}D_2 \tag{4.3}$$

$$\therefore \mathrm{RING}_R \vdash D_1 = F_{\mathrm{RING}}F_{\mathrm{ZW}}D_1 = F_{\mathrm{RING}}F_{\mathrm{ZW}}D_2 = D_2 \tag{4.4}$$

$\square$

Theorem 4.4 (RING is sound)**.** *The rules presented for* RING *are sound*

Proof. This is can be checked directly, made simpler when combined with the knowledge that the translation preserves the interpretation. $\square$

Theorem 4.5 (RING is universal)**.** RING *is universal over R-bit.*

Proof. The same interpretation-preserving translation from ZW used in Theorem 4.4 shows that any matrix expressible in ZW_R is expressible in RING_R, and since ZW_R is universal for R-bit so is RING_R. $\square$

Remark 4.6 (Comparison to Quantum Arithemtic). Quantum Arithmetic refers to performing elementary arithmetic operations to numbers stored as registers of qubits. For example one can store the number six, 110 in binary, using three qubits represented in the computational basis as:

$$|1\rangle \otimes |1\rangle \otimes |0\rangle \tag{4.5}$$

states

$$\begin{pmatrix} 1 \\ a \end{pmatrix}$$

multiplication

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

addition

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Figure 4.1: The generators of RING$_R$

Figure 4.2: Derived generators of RING$_R$: The one-sided spider for addition, the NOT gate, the multi-plication spider and the Hadamard gate .

A demanding part of Shor's algorithm [VBE96; Sho94] requires being able to perform the following elementary arithmetic operation:

$$U_{a,N} : |\,x\,\rangle \otimes |\,0\,\rangle \mapsto |\,x\,\rangle \otimes |\,a^x \mod N\,\rangle \qquad (4.6)$$

While RING is constructed from the two ring operations and compact closure, its native way of handling numbers is fundamentally different to that of quantum arithmetic. For example the state bearing the label 6 represents a single qubit, expressed as follows in the computational basis:

$$\left[\!\!\left[\begin{array}{c} | \\ \boxed{6} \end{array} \right]\!\!\right] = |\,0\,\rangle + 6|\,1\,\rangle \qquad (4.7)$$

The state representing the bit-string 110 is more complicated:

$$\left[\!\!\left[\begin{array}{ccc} \begin{array}{c} | \\ \boxed{1} \\ \boxed{-1} \\ \boxed{1} \\ \boxed{-1} \end{array} & \begin{array}{c} | \\ \boxed{1} \\ \boxed{-1} \\ \boxed{1} \\ \boxed{-1} \end{array} & \begin{array}{c} | \\ \boxed{0} \end{array} \end{array} \right]\!\!\right] = |\,1\,\rangle \otimes |\,1\,\rangle \otimes |\,0\,\rangle \qquad (4.8)$$

Since RING is universal it is still capable of representing (and reasoning about) quantum arithmetic operations, but the use of complex numbers as labels in RING should not be confused with the presentation of numbers as bitstrings in bra-ket notation.

Remark 4.7 (The power of Compact Closure). Compact Closure was a fundamental property highlighted in Ref. [AC09], the seminal work on Categorical Quantum Mechanics. That work also highlighted the importance of algebraic considerations in the design of process systems:

> In particular, we have in mind the hard-learned lessons from Computer Science of the importance of compositionality, types, abstraction, and the use of tools from algebra and logic in the design and analysis of complex informatic processes. (Categorical Quantum Mechanics, [AC09])

This new calculus demonstrates that it is possible to construct a sound, universal and complete diagrammatic calculus for quantum computing using just a single algebraic structure and compact closure. Indeed RING could be seen as further evidence of the power of compact closed structures, since without cups and caps the calculus RING is less expressive than open term-trees of rings. This idea is explored further in §5.1.

4.2   The rules of RING

We break the rules of RING into two parts. The first part are the rules relating to addition and multiplication; in particular multiplication forms a spider (the familiar Z

spider). The addition gate does not form a spider, although since it is associative and commutative on inputs we use the rounded side of the gate to indicate the inputs, and use the straight, bold edge to indicate the single output. These are found in Figure 4.3. We label these rules as simply $+$ or $\times$, since these are the effects the gates have on states.
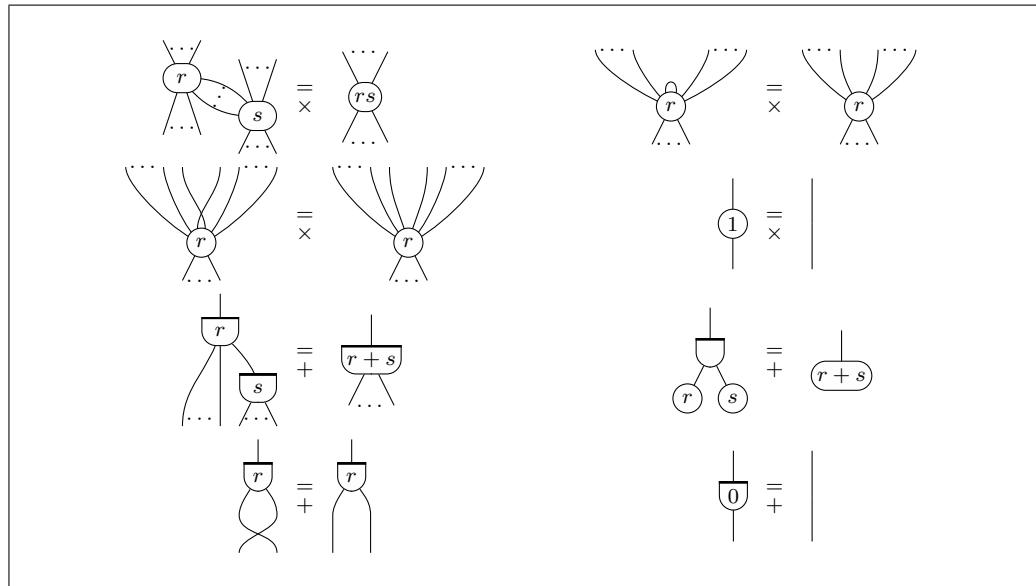
Figure 4.3: The rules of RING concerning addition and multiplication.

In Figure 4.4 we show the remaining rules of RING. These can be considered as rules containing the NOT gate (A) and (N), the distributive law (D), scalar identity (I), the Hopf law (Hopf), a law for forming a loop with addition and multiplication (L), two rules for the interactions between addition and Controlled Z (CZA1) and (CZA2), and three bialgebra laws (B1), (B2) and (CP).
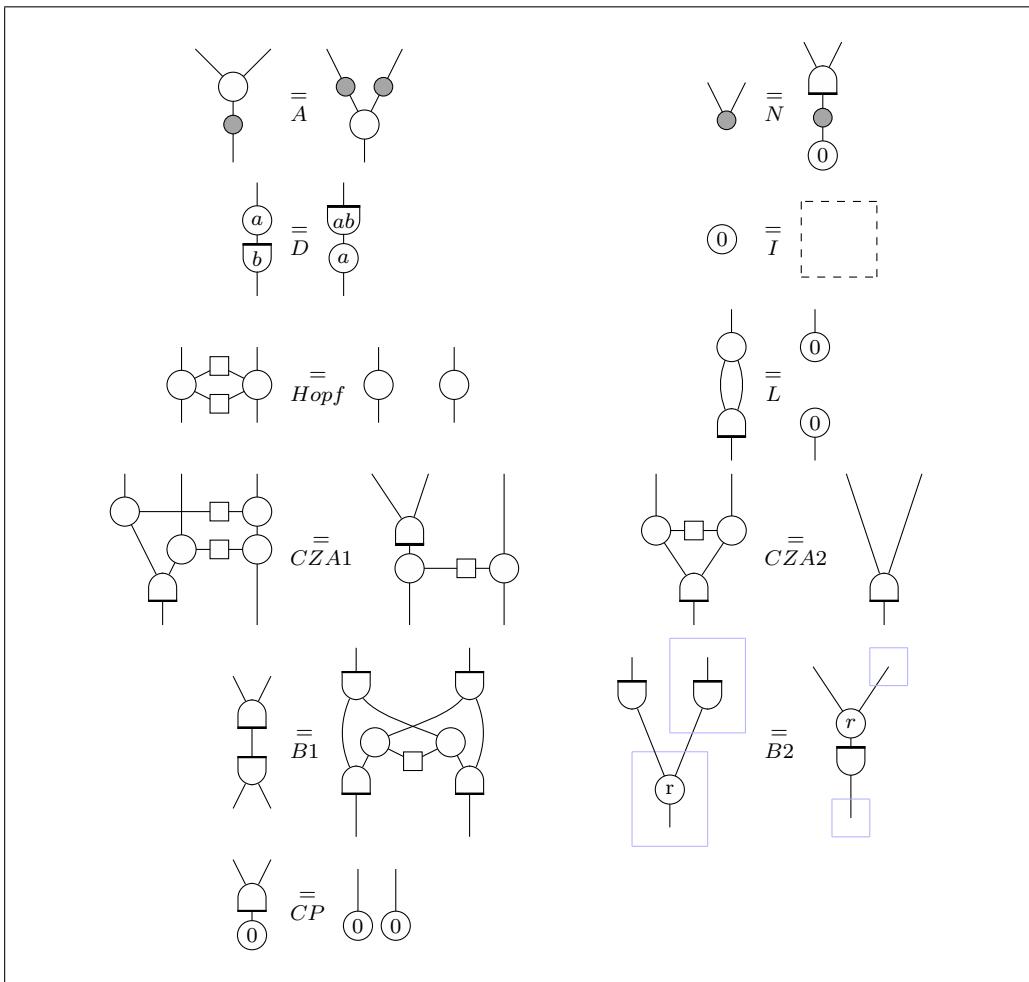
Figure 4.4: The remaining rules of RING.

The first thing we must do with our rules is justify the notational symmetry used in defining the NOT derived generator.

Lemma 4.8. *The derived generator NOT (Figure 4.2) is justified in its graphical symmetry.*

Proof.

$$(4.9)$$

□
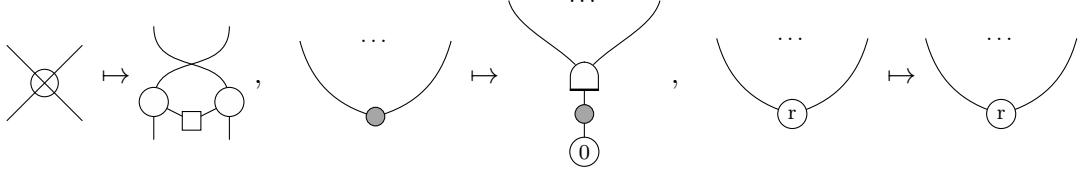
Figure 4.5: Translation from ZW_R to $\textsc{ring}_R$.

Figure 4.6: Translation from $\textsc{ring}_R$ to ZW_R

4.3   Completeness via ZW

We construct a complete ruleset for $\textsc{ring}_R$ by exhibiting a translation to and from ZW_R. This translation acts as the identity on bare wires, swaps, cups and caps. This means that the diagram components in (4.10) are considered to be the same in both calculi:

$$\left\{ \; \smile \; , \; \frown \; , \; \Box \; , \; \times \; , \; | \; \right\} \tag{4.10}$$

The translation is given in Figures 4.5 and 4.6. We will now apply the translation to each of the rules of ZW then derive these translated rules using the rules given in §4.2. In order to ease this process we shall first show that three commonly used components of ZW are translated into simple RING counterparts, namely the NOT gate, plus gate and negation. The visual similarities between the two languages can make it hard to tell at a glance which language is being considered. In this section the notation $\mapsto$ indicates the translation of a ZW diagram or equation (on the left) to a RING diagram or equation (on the right).

Lemma 4.9. *The ruleset* RING *entails that the translation of the binary w spider is the NOT gate with an input wire bent upwards:*

$$\vee \; \mapsto \; \bigcap_{0} \; = \; \vee \tag{4.11}$$

Proof. This is just the rule N □

From here on we will use Lemma 4.9 to translate the binary w spider. This NOT gate is involutive:

Lemma 4.10. *The ruleset* RING *entails the translated rule inv from ZW:*

$$\begin{array}{c}\text{(diagram)}\end{array} \tag{4.12}$$

Proof.

$$\begin{array}{c}\text{(diagram)}\end{array} \tag{4.13}$$

□

Having shown that the NOT gate is involutive we now use this lemma to show that the commonly appearing combination of a binary w spider following a larger w spider is in fact the plus gate:

Lemma 4.11. *The translation of the commonly appearing ZW diagram below is equal to the plus gate in* RING:

$$\begin{array}{c}\text{(diagram)}\end{array} \tag{4.14}$$

Proof.

$$\begin{array}{c}\text{(diagram)}\end{array} \tag{4.15}$$

$\square$

From here on we will use Lemma 4.11 to translate that composition of the $1 \to 1$ and $1 \to 2$ w spiders. To show that the traced crossing is equal to negation we will first need the following lemmas:

Lemma 4.12. *The following equations are derivable in* RING*:*

$$(4.16)$$

Proof.

$$(4.17)$$

$$(4.18)$$

$\square$

Lemma 4.13. *The following equation is derivable in* RING*:*

$$(4.19)$$

Proof.

$$(4.20)$$

$$\underset{+,A}{=} \qquad = \qquad \underset{inv,\times,+}{=} \qquad \boxed{\Pi\, a_1\ldots a_n} \qquad (4.21)$$

$\square$

Lemma 4.14. *The ruleset* RING *entails the translation of the traced crossing is multiplication by* -1:

$$\mapsto \qquad = \qquad (4.22)$$

Proof.

$$\underset{\times}{=} \qquad = \qquad \underset{\times}{=} \qquad \underset{4.13}{=} \qquad \underset{+,\times}{=} \qquad (4.23)$$

$\square$

From here on we will use Lemma 4.14 to translate the traced crossing from ZW to RING. Having derived these implicit simplifications in the translation we will now translate and derive the rules of ZW in the order presented in [Had17], with the exception of the *inv* rule, which was shown in Lemma 4.10.

Lemma 4.15. *The ruleset* RING *entails the wire bending rules of ZW:*

$$\qquad = \qquad = \qquad \qquad = \qquad \qquad = \qquad (4.24)$$

Proof. This is just the requirement that RING_R is compact closed. $\square$

Lemma 4.16. *The ruleset* RING *entails the translated rule* rei_2^x *from ZW:*

$$\qquad = \qquad \qquad \mapsto \qquad = \qquad \qquad (4.25)$$

Proof. Apply rules $\times$ and $Hopf$.   □

Lemma 4.17. *The ruleset* RING *entails the translated rule* rei_3^x *from ZW:*

$$\tag{4.26}$$

Proof. After applying rule $\times$ both sides are isometric to:

$$\tag{4.27}$$

□

Lemma 4.18. *The ruleset* RING *entails the translated rule* rei_1^x *from ZW:*

$$\tag{4.28}$$

Proof. Tautological, thanks to Lemma 4.14.   □

Lemma 4.19. *The ruleset* RING *entails the translated rule* nat_x^η *from ZW:*

$$\tag{4.29}$$

Proof. This is identical to rei_x^2 from Lemma 4.16.   □

Lemma 4.20. *The ruleset* RING *entails the translated rule* nat_x^ϵ *from ZW:*

$$\tag{4.30}$$

Proof. This is identical to rei_x^2 from Lemma 4.16.   □

Lemma 4.21. *The ruleset* RING *entails the translated rule* nat_x^w *from ZW:*

$$(4.31)$$

Proof. This is just rule CZA1.   □

Lemma 4.22. *The ruleset* RING *entails the translated rule* cut_w *from ZW:*

$$(4.32)$$

Proof.

$$(4.33)$$

□

The next rule to translate (Lemma 4.24) requires us to be able to apply a cup as an input to the plus gate. We will first provide a quick lemma to show that this results in the zero state:

Lemma 4.23. *The following is derivable in* RING:

$$(4.34)$$

Proof.

$$(4.35)$$

□

Lemma 4.24. *The ruleset* RING *entails the translated rule* tr_w *from ZW:*

(4.36)

Proof.

(4.37)

□

Lemma 4.25. *The ruleset* RING *entails the translated rule* sym_w *from ZW:*

(4.38)

Proof. This is symmetry of addition in $+$.   □

The rule ba_w from [Had17] is, frankly, too visually complicated once translated. Instead we rely on the fact that ba_w can be derived from $2a$, $5a$, $5b$ and $5c$ of [Had15] as shown in [Had15, proposition 4]. Rule $2a$ of Ref. [Had15] happens to be the same as the rule *inv* of Ref. [Had17], and the translation of *inv* was already derived in Lemma 4.10. The remaining three rules we modify slightly and take as axioms B1, CP and I in RING.

Lemma 4.26. *The translations of three rules* $5a$, $5b$ *and* $5c$ *of Ref. [Had15] into* RING *are derivable:*

(4.39)

$$\text{(4.40)}$$

$$\text{(4.41)}$$

Proof.

$$\text{(4.42)}$$

$$\text{(4.43)}$$

$$\text{(4.44)}$$

$\square$

Before our next derivation we need a lemma that shows the interaction between the NOT and H derived generators.

Lemma 4.27. *The composition of the NOT and H derived generators in* RING *gives the equality:*

$$\text{(4.45)}$$

Proof.

$$\tag{4.46}$$

$$\square$$

Lemma 4.28. *The ruleset* RING *entails the translated rule* ant_x^n *from ZW:*

$$\tag{4.47}$$

Proof.

$$\tag{4.48}$$

$$\square$$

Lemma 4.29. *The following rules are translated verbatim from ZW, and are all derivable in* RING*:*

$$\tag{4.49}$$

$$\tag{4.50}$$

Proof. These are all derivable from $\times$. $\square$

Lemma 4.30. *The ruleset* RING *entails the translated rule* ba_{zw} *from ZW, noting there*

must be at least one output (boundary at the top of the diagram):

$$(4.51)$$

Proof. This is rule $B2$ □

Lemma 4.31. *The ruleset* RING *entails the translated rule loop from ZW:*

$$(4.52)$$

Proof. This is rule L and $\times$.   □

Lemma 4.32. *The ruleset* RING *entails the translated rule ph from ZW:*

$$(4.53)$$

Proof. This is spider fusion and un-fusion from rule $\times$.   □

Lemma 4.33. *The ruleset* RING *entails the translated rule* nat_c^n *from ZW:*

$$(4.54)$$

Proof. This is rule A.   □

Lemma 4.34. *The ruleset* RING *entails the translated rule* unx *from ZW:*

$$(4.55)$$

Proof.

$$(4.56)$$

□

Lemma 4.35. *The ruleset* RING *entails the translated rule* rng_{-1} *from ZW:*

$$(4.57)$$

Proof. This is tautological, because of Lemma 4.14.   □

Lemma 4.36. *The ruleset* RING *entails the translated rule* $rng_+^{r;s}$ *from ZW:*

$$(4.58)$$

Proof.

$$(4.59)$$

$\square$

Lemma 4.37. *The equality* $[\![\, [\![\, g \,]\!]_{ZW} \,]\!]_{\mathrm{RING}_R} = g$ *is derivable in* RING *for each generator* g.

Proof.

$$\left[\!\!\left[\,\left[\!\!\left[\,\raisebox{-0.5em}{\includegraphics{}}\,\right]\!\!\right]_{ZW}\,\right]\!\!\right]_{\mathrm{RING}_R} = \left[\!\!\left[\,\raisebox{-0.5em}{\includegraphics{}}\,\right]\!\!\right]_{\mathrm{RING}_R} = \raisebox{-0.5em}{\includegraphics{}} \tag{4.60}$$

$$\left[\!\!\left[\,\left[\!\!\left[\,\raisebox{-0.5em}{\includegraphics{}}\,\right]\!\!\right]_{ZW}\,\right]\!\!\right]_{\mathrm{RING}_R} = \left[\!\!\left[\,\raisebox{-0.5em}{\includegraphics{}}\,\right]\!\!\right]_{\mathrm{RING}_R} \underset{4.11}{=} \raisebox{-0.5em}{\includegraphics{}} \tag{4.61}$$

$$\left[\!\!\left[\,\left[\!\!\left[\,\raisebox{-0.5em}{\includegraphics{}}_{a}\,\right]\!\!\right]_{ZW}\,\right]\!\!\right]_{\mathrm{RING}_R} = \left[\!\!\left[\,\raisebox{-0.5em}{\includegraphics{}}_{a}\,\right]\!\!\right]_{\mathrm{RING}_R} = \raisebox{-0.5em}{\includegraphics{}}_{a} \tag{4.62}$$

$\square$

This concludes our collection of proofs showing that RING_R is complete.

4.4   Extending ZH, and relating it to RING

The calculus $\mathrm{ZH}_{\mathbb{C}}$ was shown to be complete via the existence of its own normal form, rather than via equivalence with ZW [BK19]. In this section will give a generalisation of ZH, called ZH_R, and we will then exhibit translations between ZH_R and RING_R. It must be noted that while ZW_R exists and is well defined for any commutative ring R, the calculus ZH_R is only defined for commutative rings R with a half.

Although we have already provided a complete ruleset for RING_R from ZW_R in §4.2, ZW and ZH have very different rulesets of their own [BK19; Had17]. The translations of these rulesets into RING remain distinct, and therefore one may be more useful than the other in different situations. The translation from RING_R to ZH_R was rather easy to find; one simply had to find a ring substructure in ZH (an idea continued in §5), which required looking at the rules of ZH to see where addition and multiplication were used. Translating from ZH to RING_R was similarly easy; since we have access to elementary matrices it was simple enough to find a translation for low-arity H-boxes, and then generalise from there.

Since ZH_R has the additional requirement on R that it be a commutative ring with $\frac{1}{2}$ (compared to ZW_R, where R simply needs to be a commutative ring) we simply state here a smaller ruleset (§4.4.3) for RING and translation to and from ZH_R (Figure 4.9), relegating the proof of completeness to Appendix B.

4.4.1   The graphical calculus ZH_R

The graphical calculus ZH_R is almost identical to that of $\mathrm{ZH}_{\mathbb{C}}$, with the exception that the labels of the H-boxes are now from some arbitrary commutative ring with a half, R. Explicitly this means that R has a characteristic other than 2, and that there is an element $\frac{1}{2}$ which is the multiplicative inverse of 2.

Definition 4.38 (Generalised ZH)**.** ZH_R has generators and interpretation as given in Figure 4.7, and a ruleset given in Figure 4.8.

Theorem 4.39 (ZH_R is complete)**.** *The rules presented in Figure 4.8, are sound and complete for the universal calculus ZH_R, where R is a commutative ring with a half.*

Proof. The proof is given in Appendix A, because it follows an almost identical proof to that of [BK19]. □

Theorem 4.40 (RING is complete, via ZH_R)**.** *The rules of §4.4.3 are complete for the graphical calculus RING_R, where R is a commutative ring with a half.*

Proof. The proof is given in Appendix B. □

Z spiders $Z : m \to n$ $|0\rangle^{\otimes n}\langle 0|^{\otimes m} + |1\rangle^{\otimes n}\langle 1|^{\otimes m}$

H-boxes $H_r : m \to n$ $\sum r^{i_1\dots i_m j_1\dots j_n}|j_1\dots j_n\rangle\langle i_1\dots i_m|$

Figure 4.7: The generators and interpretation of ZH_R
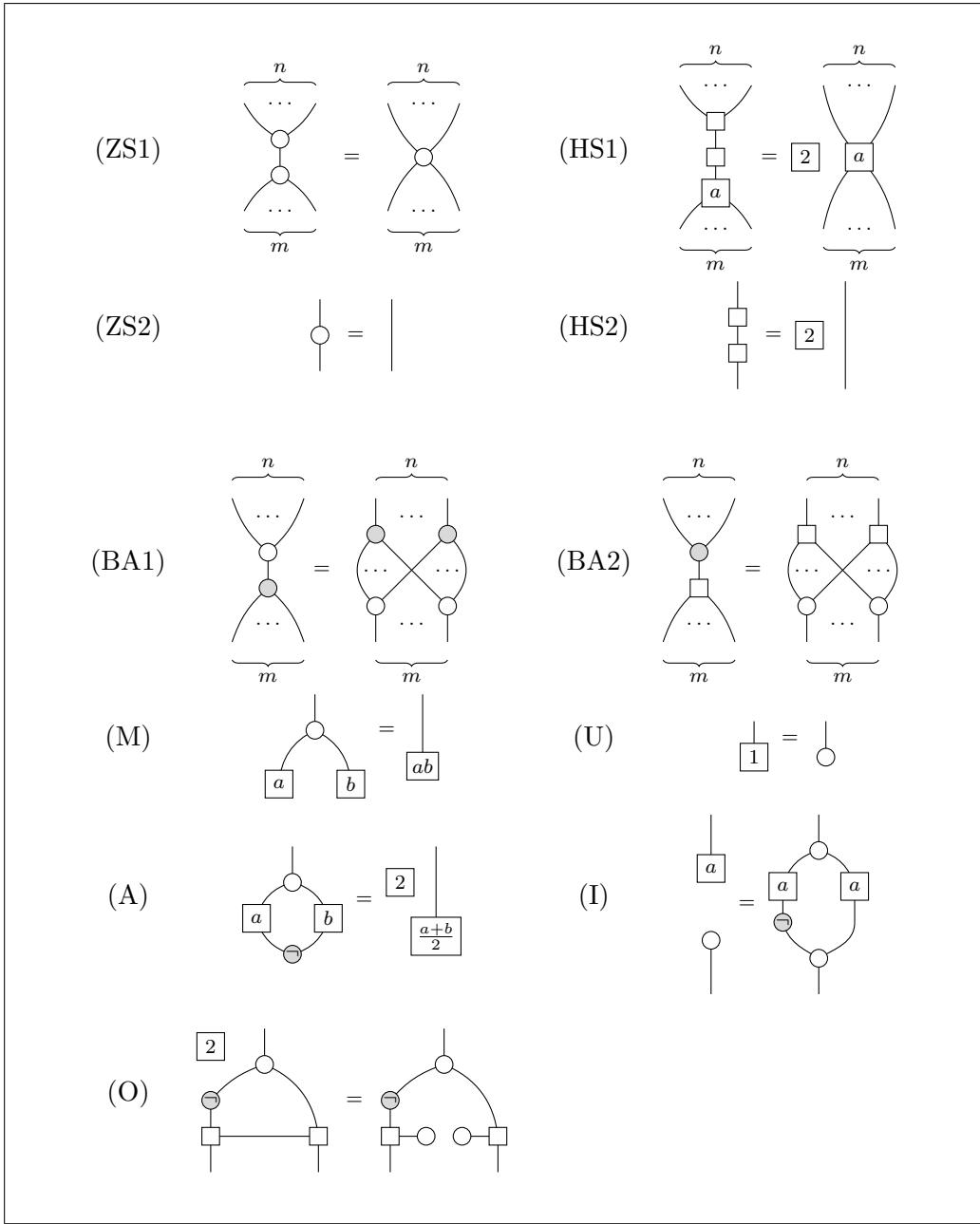
Figure 4.8: The rules for ZH_R, note that these are almost identical to the rules for $\mathrm{ZH}_\mathbb{C}$ as presented in [BK19]. Throughout, m, n are non-negative integers and in the original paper a, b were arbitrary complex numbers. In ZH_R we allow a, b to be arbitrary elements of the ring R, where R is commutative with a half. The right-hand sides of both *bialgebra* rules (BA1) and (BA2) are complete bipartite graphs on $(m + n)$ vertices, with an additional input or output for each vertex. The horizontal edges in equation (O) are well-defined because only the topology matters and we do not need to distinguish between inputs and outputs of generators. The rules (M), (A), (U), (I), and (O) are pronounced *multiply*, *average*, *unit*, *intro*, and *ortho*, respectively.

4.4.2   Translation between RING and ZH

The translation between $\textsc{ring}_R$ and ZH_R is given as Figure 4.9. This translation preserves the interpretation into R-bit.

ZH$_R \to$ RING$_R$

RING$_R \to$ ZH$_R$
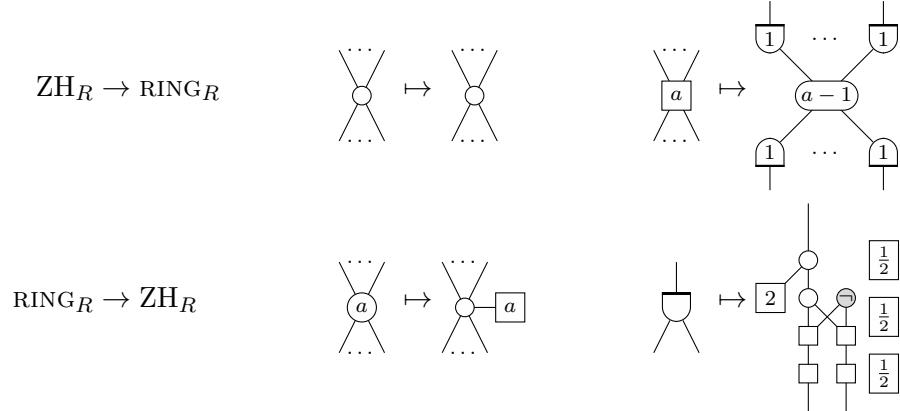
Figure 4.9:   Translations between $\textsc{ring}_R$ and ZH_R

4.4.3   A small, complete ruleset of RING from ZH

This ruleset is the result of translating the rules from ZH into $\textsc{ring}$. We have taken some effort to remove redundant rules from the list below, although we do not make any claims as to necessity of individual rules. This list is included to provide a contrast against the rules of §4.2. We have also left the redundant ring rules in since they are a foundational aspect of $\textsc{ring}_R$. We present these somewhat reduced rules here, divided into sections:

- Ring Rules. These rules have clear interpretations as the ring operations in $\textsc{ring}_R$.
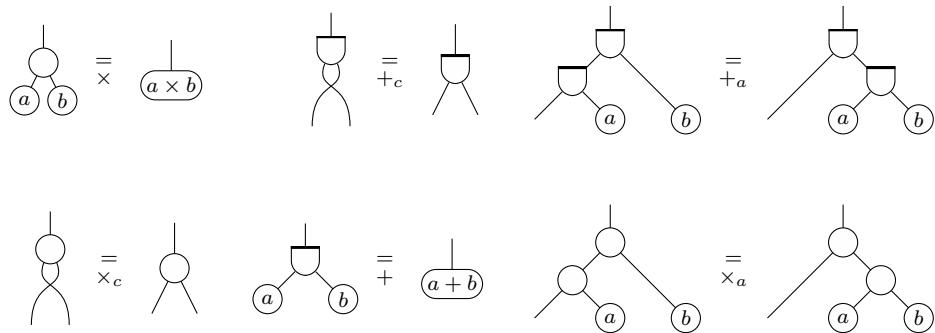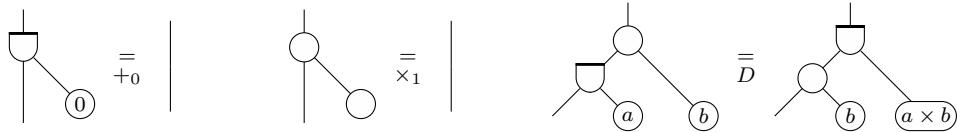
- The multiplication spider, scalar multiplication, and Hadamard. As always the diagonal dots indicate one or more wires, and the horizontal dots indicate zero or more wires.

$$(4.63)$$

- Bialgebra rules.

$$(4.64)$$
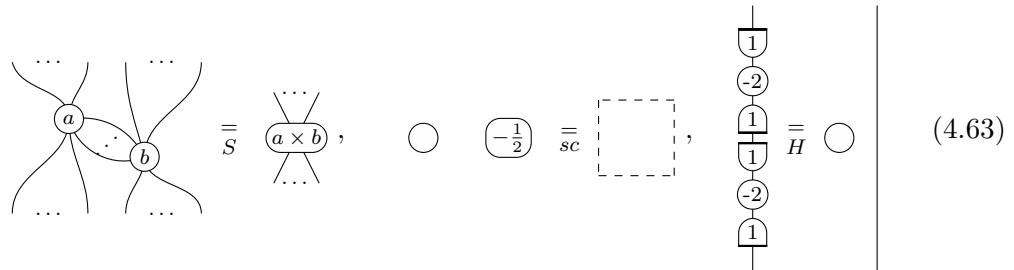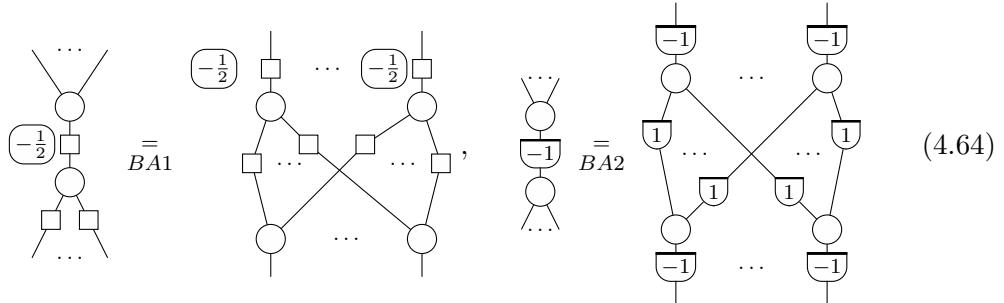
- The *intro* and *ortho* rules.

$$(4.65)$$
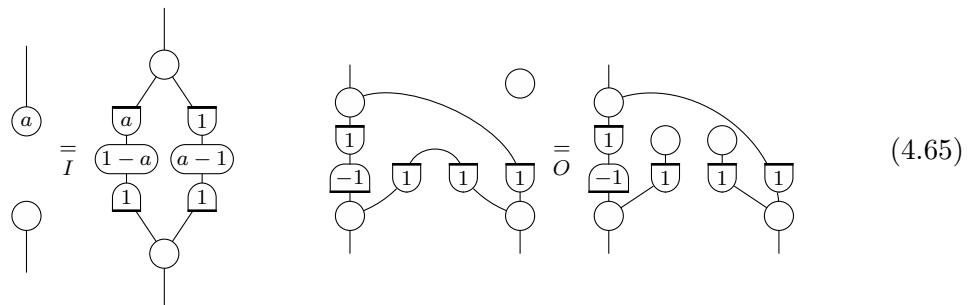
- Finally the rule that links the plus gate with its image after being translated into

ZH$_R$ and back again.

$$\text{(4.66)}$$

4.5   Summary

In our first results chapter we have introduced the new graphical calculus RING, showed its completeness using ZW, and extended ZH. RING itself is generated only by ring operations (acting on states) and compact closed structure. This sparsity of generators will allow us to use RING as a 'generic' phase ring calculus in the next chapter, and so investigate the limitations and uses of phase ring graphical calculi. The focus of the next chapter is on the category of phase ring graphical calculi, more precisely the category of graphical calculi that model ring structure in the manner of ZW, ZH and RING, and the morphisms that preserve soundness.

Chapter 5

Phase Ring Graphical Calculi and Phase Homomorphisms

In this chapter:

> ▷ We describe the category of phase ring graphical calculi

> ▷ We justify RING_R being 'initial, up to ring isomorphism'

> ▷ We justify RING_K being 'the generic phase field calculus'

> ▷ We investigate when algebra homomorphisms lift to soundness- and proof- preserving maps

> ▷ This chapter requires knowledge of the calculus RING from §4, and the idea of phase algebra homomorphisms will be key to parts of §7 and §8.

The calculus RING_R, introduced in §4, resulted from investigating the topics we shall cover in this chapter. We would like to know what extra structure we achieve by describing a calculus as having a phase ring (or phase algebra in general) over just a set of labels. We shall phrase this as a category of phase ring graphical calculi, and then investigate the morphisms of this category that arise from morphisms of rings. We will only be looking at phase rings that act as monoids on states of arity $0 \to 1$, i.e. situations where elements of arity $0 \to 1$ act like elements in a ring, with some gates of arity $2 \to 1$ acting like monoids on those states. This is how ZW and ZH act as phase rings, with ZX acting analogously but on a phase group instead. The phase group calculus ZQ (§6) does not act in this manner, for reasons covered in that chapter.

In §5.1 we look at the category of Phase Ring Graphical Calculi as Monoids before exhibiting the calculi RING_R, ZW_R and ZH_R as objects in this category. This is compared to the earlier work performed in Ref. [DK13], which also looked at models of algebraic structures in graphical calculi. We then investigate how ring homomorphisms $\phi : S \to S'$

relate to morphisms out of RING$_S$ in this category, laying the groundwork for §5.2 and §5.3.

In §5.2 we look at the calculus RING$_K$ where K is a field, showing that RING$_K$ has an essential uniqueness in its interpretation, as well as the property of being 'initial, up to field automorphism' for K-phase-field calculi into K-bit. In §5.3 we look at how ring morphisms can lift to morphisms of certain graphical calculi, and how ring homomorphisms can lift to soundness- and proof- preserving maps of theorems. Finally in §5.4 we give a similar treatment to phase *group* homomorphisms, classifying the phase group endomorphisms available in certain finite fragments of ZX, and then show how to generalise these ideas to Σ-algebras.

We derive these results in the generality afforded to us by the ring structure of both the phase ring and the category R-bit. For quantum graphical calculi, however, we only need to work over $\mathbb{C}$-bit, and we shall show later on that this severely restricts the phase ring structure of the graphical calculus as well. The effect of this is that the calculus RING$_\mathbb{C}$ will embed naturally into any universal qubit graphical calculus with a phase ring. RING$_\mathbb{C}$ therefore acts as a common (universal and complete) core for all phase ring qubit graphical calculi.

5.1   The category of phase ring graphical calculi

We begin this chapter with an intention of finding a 'generic' phase ring calculus. In order to tackle this problem we build on the idea of models of PROPs from Ref. [Mac65] and introduce a category of Phase Ring Graphical Calculi (Definition 5.5). From there we show that RING$_R$ has certain desirable properties, such as the fact that given any graphical calculus G with phase ring structure R there is exactly one map from RING$_R$ to G that preserves that structure (Corollary 5.14).

The author did not feel they could justify calling RING$_R$ 'the' generic phase ring calculus; there are too many ways to model phase ring structure over R-bits. They do, however, feel justified in calling RING$_K$ 'the' generic phase field calculus over K-bit for reasons that we show in §5.2. The idea of a phase ring homomorphism that we begin in this section (Definition 5.16) is continued in §5.3, and then generalised to other algebras in §5.4.

5.1.1   Objects in Ring-GC (R)

We begin with as generic a notion of phase ring structure as we can. Consider a commutative ring which is a set S equipped with multiplication $\times : 2 \to 1$, addition $+ : 2 \to 1$, and with the distinguished elements 0 and 1.

Definition 5.1 (RING-PROP$_S$)**.** We construct our generic ring PROP RING-PROP$_S$ for

the ring $(S, +, \times)$ with the following generators:

$$\forall s \in S \text{ the element} \qquad \underset{\textstyle s}{|} \qquad\qquad 0 \to 1 \qquad\qquad (5.1)$$

$$\text{addition gate} \qquad \boxed{+} \qquad\qquad 2 \to 1 \qquad\qquad (5.2)$$

$$\text{multiplication gate} \qquad \boxed{\times} \qquad\qquad 2 \to 1 \qquad\qquad (5.3)$$

We then quotient out these two rewrite rules:

$$\boxed{\times}_{a\ b} \overset{=}{\underset{\text{multiplication}}{}} (a \times b) \qquad\qquad \boxed{+}_{a\ b} \overset{=}{\underset{\text{addition}}{}} (a + b) \qquad (5.4)$$

There is a category of all such PROPs, defined by being the image of the functor from the category of rings that sends S to RING-PROP$_S$, and sends a ring homomorphism ϕ to a strict symmetric monoidal functor $\bar{\phi}$:

$$\bar{\phi} : \text{RING-PROP}_S \to \text{RING-PROP}_{S'} \qquad\qquad (5.5)$$

$$\underset{\textstyle s}{|} \mapsto \underset{\textstyle \phi(s)}{|} \qquad\qquad \boxed{+} \mapsto \boxed{+} \qquad \boxed{\times} \mapsto \boxed{\times} \qquad (5.6)$$

Remark 5.2. These two rewrite rules in (5.4) imply that our addition and multiplication gates are commutative and associative *when applied directly to states*, purely because those properties are already present in the ring S. For example:

$$\boxed{\times}_{a\ b} \overset{=}{\underset{\text{multiplication}}{}} (a \times b) = (b \times a) \overset{=}{\underset{\text{multiplication}}{}} \boxed{\times}_{b\ a} \qquad (5.7)$$

This is a weaker property than saying that these gates are commutative etc. in general.

Definition 5.3 (Models of RING-PROP). A model for RING-PROP$_S$ in some graphical calculus G is a morphism of PROPs RING-PROP$_S \to G$.

A model of RING-PROP$_S$ into a graphical calculus can be seen as choosing diagrams $\{g_s\}$, g_+ and $g_\times$ to represent states, an addition gate, and a multiplication gate with the syntactic behaviour suggested by their names.

Example 5.4 (Degenerate model). We construct the graphical calculus ZERO as having generators O_m^n, one for each arity $m \to n$, interpreted as the zero matrix of size $2^m \to 2^n$,

and the following rewrite rules:

$$O_m^n \circ O_l^m = O_l^n \tag{5.8}$$

$$O_a^b \otimes O_c^d = O_{a+c}^{b+d} \tag{5.9}$$

For any PROP $\mathbb{P}$ there is a strict symmetric monoidal functor from $\mathbb{P}$ to ZERO; in particular there is a unique model of RING-PROP$_S$ in ZERO for any S, but it is not faithful.

Definition 5.5 (Ring-GC(R)). The category of Phase Ring Graphical Calculi (as monoids) into R-bit, shortened to **Ring-GC(R)**, has as objects graphical calculi over R-bit that contain a model of RING-PROP$_S$ for some S, and morphisms that preserve this RING-PROP$_S$ structure. Specifically:

- Objects are pairs $(G, \mathcal{M}_S)$, where G is a (compact closed) graphical calculus given as a collection of generators, an interpretation $[\![\cdot]\!]_G$, and rewrite rules, and $\mathcal{M}_S$ is a model of RING-PROP$_S$ in G.

- A morphism

$$f : (G, \mathcal{M}_S) \to (G', \mathcal{M}_{S'}) \tag{5.10}$$

  is a strict symmetric monoidal functor $f_\mathbb{D} : G \to G'$ and also a ring homomorphism $f_\mathbb{R} : S \to S'$ such that the right hand diagram commutes:

$$
\begin{array}{ccccc}
S & & \text{RING-PROP}_S & \xrightarrow{\mathcal{M}_S} & G \\
\downarrow{f_\mathbb{R}} & & \downarrow{\bar{f}_\mathbb{R}} & & \downarrow{f_\mathbb{D}} \\
S' & & \text{RING-PROP}_{S'} & \xrightarrow{\mathcal{M}_{S'}} & G'
\end{array}
\tag{5.11}
$$

Proposition 5.6. *The above definition of* **Ring-GC(R)** *is indeed a category.*

Proof. We just need to show the following properties:

- Identity morphisms f are given by the identity morphism $f_\mathbb{D}$ of compact closed PROPs and $f_\mathbb{R}$ of commutative rings respectively, and this satisfies the condition of preserving the distinguished diagrams $\{g_s\}, g_+, g_\times$

- Composition of morphisms exists for $f_\mathbb{D}$ and $f_\mathbb{R}$, and the commutative diagram of (5.11) composes vertically

- Associativity of composition of morphisms follows from the associativity of composition of the functors $f_\mathbb{D}$ and the ring homomorphisms $f_\mathbb{R}$

$\square$

Remark 5.7. In this thesis we are using a *syntactic* version of the models of RING-PROP$_S$. In Definition 5.1, where we stated that addition and multiplication should be rules (i.e. equivalences) in RING-PROP$_S$ we could instead have given a requirement on the interpretation of any model, such as:

$$\left[\!\left[\mathcal{M}_S \left(\begin{array}{c} \times \\ \widehat{a} \;\; \widehat{b} \end{array} \right) \right]\!\right] = \left[\!\left[\mathcal{M}_S \left(\widehat{a \times b} \right) \right]\!\right] \tag{5.12}$$

$$\left[\!\left[\mathcal{M}_S \left(\begin{array}{c} + \\ \widehat{a} \;\; \widehat{b} \end{array} \right) \right]\!\right] = \left[\!\left[\mathcal{M}_S \left(\widehat{a + b} \right) \right]\!\right] \tag{5.13}$$

The author feels that this semantic version better reflects the way these calculi are researched: First by constructing generators with certain interpretations and then finding rules. The syntactic version we use, however, is far neater. All the results in this chapter can be adapted to semantic versions with little effort.

Definition 5.8. We will regard the calculus RING$_S$ as being an object of **Ring-GC** (R) whenever S is a subring of R, exhibited as RING$_S$ with interpretation $[\![\cdot]\!]$: RING$_S \to$ S-bit naturally extending to $[\![\cdot]\!]$: RING$_S \to R$-bit, and with model $\mathcal{M}_S$ the identification of the ring structure inherent in RING$_S$:

$$\begin{array}{c} \mid \\ \widehat{s} \end{array} \mapsto \begin{array}{c} \mid \\ \widehat{s} \end{array} \qquad\qquad \begin{array}{c} + \\ \end{array} \mapsto \begin{array}{c} \cup \\ \end{array} \qquad\qquad \begin{array}{c} \times \\ \end{array} \mapsto \begin{array}{c} \bigcirc \\ \end{array} \tag{5.14}$$

The addition and multiplication rules of RING$_S$ are the image of the addition and multiplication rules of RING-PROP$_S$ under $\mathcal{M}_S$.

Remark 5.9. Definition 5.8 shows that whenever $S \subset R$ we have a model of RING-PROP$_S$ in **Ring-GC** $(R$-bit$)$. It could well be the case that there are faithful models of some RING-PROP$_S$ in **Ring-GC** $(R$-bit$)$ where $R \subsetneq S$, but we will show in §5.2 that this cannot be the case when R is a field.

Lemma 5.10. *The calculi ZW$_S$ and ZH$_S$ (with the usual phase ring structure) can be exhibited as objects of* **Ring-GC** $(R$-bit$)$ *whenever S is a subring of R*

Proof. For ZW$_S$ we extend its interpretation to be into R-bit, and use the model $\mathcal{M}_S$ given by:

$$\mathcal{M}_S : \text{RING-PROP}_S \to \text{ZW}_S \tag{5.15}$$

$$\begin{array}{c} \mid \\ \widehat{s} \end{array} \mapsto \begin{array}{c} \mid \\ \widehat{s} \end{array} \qquad \forall s \in S \tag{5.16}$$

(5.17)

(5.18)

Likewise for ZH_S we extend the interpretation and use the model given by:

$$\mathcal{M}_S : \text{RING-PROP}_S \to \mathrm{ZH}_S \tag{5.19}$$

 $\qquad \forall s \in S$

(5.20)

(5.21)

(5.22)

Since these are the 'obvious' models of RING-PROP$_S$ suggested by the addition and multiplication rules of ZW and ZH we shall simply refer to these objects of **Ring-GC** (R-bit) as ZW_S and ZH_S. The requirements of Definition 5.5 are shown by checking:

(5.23)

(5.24)

$\square$

Definition 5.11 (Generic object in **Ring-GC** (R-bit)). We will simply write G_S to indicate a generic object $(G, \mathcal{M}_S)$ in **Ring-GC** (R-bit). We will also distinguish the

image of the generators of RING-PROP$_S$ under $\mathcal{M}_S$ as:

$$g_s := \mathcal{M}_S \left(\underset{s}{\downarrow} \right) \qquad g_+ := \mathcal{M}_S \left(\boxed{+} \right) \qquad g_\times := \mathcal{M}_S \left(\boxed{\times} \right) \qquad (5.25)$$

The idea behind the definition we give for **Ring-GC** (R-bit) is similar to that of MacLane's definition of algebras for a PROP (Definition 5.12). In the usual definition a PROP was held fixed and all possible models of that PROP were considered. In this work we want to be able to consider multiple phase rings (i.e. a related family of PROPs) while also restricting the models into only graphical calculi over R-bit, and only those models that are the identity on objects.

Definition 5.12 (Algebras for a PROP [Mac65, §5]). For a PROP $\mathbb{P}$ representing an algebraic theory $\mathcal{A}$ an *algebra of type* $\mathcal{A}$ is a product-preserving functor from $\mathbb{P}$ to the category of sets.

5.1.2   Some morphisms in Ring-GC (R)

Now that we know the objects in this category, let us give some example morphisms. Of the phase ring graphical calculi we have discussed RING$_S$ has the least 'extra' structure beyond that which is necessary for phase rings. It is perhaps unsurprising that the morphisms out of RING$_S$ are determined by the ring morphisms out of S.

Proposition 5.13 (Morphisms out of RING$_S$).

$$Hom_{\textbf{Ring-GC}\,(R\text{-}bit)}[\text{RING}_S, G_{S'}] \cong Hom_{Ring}[S, S'] \qquad (5.26)$$

Proof. We first note that every map ϕ in $\text{Hom}_{\text{Ring}}[S, S']$ lifts to the map

$$\text{RING}_S \xrightarrow{(\phi_\mathbb{D}, \phi)} G_{S'} \qquad (5.27)$$

where

$$\phi_\mathbb{D} : \text{RING}_S \to G \qquad (5.28)$$

$$\underset{a}{\downarrow} \mapsto g_{\phi(a)} \qquad \boxed{+} \mapsto g_+ \qquad \boxed{\times} \mapsto g_\times \qquad (5.29)$$

which satisfies the conditions of Definition 5.5. Next we note that for a generic morphism $f = (f_\mathbb{D}, f_\mathbb{R})$ of $\text{Hom}_{\textbf{Ring-GC}\,(R\text{-bit})}[\text{RING}_S, G_{S'}]$, the diagram map $f_\mathbb{D}$ is uniquely determined by $f_\mathbb{R}$, because RING$_S$ is generated by

$$\underset{a}{\downarrow} \,, \quad \boxed{+} \,, \quad \boxed{\times} \,, \qquad (5.30)$$

and the conditions of Definition 5.5 determine $f_{\mathbb{D}}$ as:

$$f_{\mathbb{D}} : \text{RING}_S \to G \tag{5.31}$$

 $\mapsto g_{f_{\mathbb{R}}(s)}$ $\mapsto g_+$ $\mapsto g_\times$ (5.32)

Therefore every map out of RING_S in **Ring-GC** (R-bit) is of the form $f = (\phi_{\mathbb{D}}, \phi)$. We therefore have the equivalence:

$$\text{Hom}_{\textbf{Ring-GC}\,(R\text{-bit})}[\text{RING}_S, G_{S'}] \cong \text{Hom}_{\text{Ring}}[S, S'] \tag{5.33}$$

$$(\phi_{\mathbb{D}}, \phi) \leftrightarrow \phi \tag{5.34}$$

$\square$

Suppose that instead of just looking at morphisms out of RING_R we also wanted to look at morphisms into some generic object G_S that preserved ring structure.

Corollary 5.14. *Given a generic object $G_{S'}$ and an isomorphism $\phi : S \cong S'$ there is exactly one morphism $(f_{\mathbb{D}}, f_{\mathbb{R}})$ from RING_S to $G_{S'}$ such that $f_{\mathbb{R}} = \phi$*

Remark 5.15. Corollary 5.14 is the justification for saying that RING_S is 'initial for graphical calculi with phase ring S'. The degenerate example ZERO can be seen as similarly terminal; for any ring S there is a unique model $\mathcal{M}_S$ of RING-PROP_S in ZERO, and for any object G_S of **Ring-GC** (R-bit) there is a unique morphism from G_S to $(\text{ZERO}, \mathcal{M}_S)$ formed by sending every diagram in G_S to the unique diagram of the same arity in ZERO.

Another example of a morphism in **Ring-GC** (R-bit) is that of the phase ring homomorphism we are about to define. Phase ring homomorphisms are in essence the lifting of a ring homomorphism $\phi : S \to S'$ not to $\bar{\phi} : \text{RING-PROP}_S \to \text{RING-PROP}_{S'}$ but to a functor between two members of a parameterised family of calculi (such as $\text{RING}_S \to \text{RING}_{S'}$). Such a lifting is not guaranteed to exist, and depends on both the calculus and the model. We will define these phase ring homomorphisms for RING_S, ZH_S and ZW_S in §5.3, but the definition relies on the way phases are presented in the calculus.

Definition 5.16 (Phase ring homomorphism for RING). Given a ring homomorphism $\phi : S \to S'$, we can lift ϕ to the symmetric monoidal functor $\hat{\phi} : \text{RING}_S \to \text{RING}_{S'}$ where we let ϕ act on each phase:

 $\mapsto$ $\mapsto$ $\mapsto$ (5.35)

This $\hat{\phi}$ is called a 'phase ring homomorphism', with the pair $(\hat{\phi}, \phi)$ being a morphism in **Ring-GC** (R-bit). There is a similar lifting for semantics, where ϕ is lifted to the

symmetric monoidal functor $\tilde{\phi} : S\text{-bits} \to S'\text{-bits}$ which is the identity on objects, and acts on matrices as:

$$
\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & & \end{pmatrix} \mapsto \begin{pmatrix} \phi(a_{11}) & \phi(a_{12}) & \dots \\ \phi(a_{21}) & \phi(a_{22}) & \dots \\ \vdots & & \end{pmatrix} \tag{5.36}
$$

We touch on this structure of $(\phi, \hat{\phi}, \tilde{\phi})$ again in §5.3.

Note that in **Ring-GC** (R) we have no restrictions on the ring substructures we are modelling with RING-PROP$_S$. There is also no interplay between the interpretation $[\![\, \cdot \,]\!]$ and the ring structure. In the next subsection we limit ourselves to **Ring-GC** (K), for K a field, and show the very strict requirements that result from *faithful* interpretations of RING-PROP$_S$. After we have examined fields we shall return to the idea of a phase ring homomorphism.

5.2   The generic phase field calculus

In this section K is a field, and R is a commutative ring. Faithfulness of the interpretation for RING-PROP$_R$ (Definition 5.17) not only allows us to avoid degenerate cases, but also imposes profound restrictions on the interpretation which we will cover in this section. We first note that the restriction forces two important properties for the states $\overset{\shortmid}{(0)}$ and $\overset{\shortmid}{(1)}$; in particular that they are both non-zero, and non-colinear, and so form a basis for K^2 (Lemma 5.21) which in turn forces the interpretation of the other generators of RING-PROP$_R$ (Lemma 5.23). From there we show several properties of RING$_K$, which we summarise in Remark 5.36.

Definition 5.17 (Faithful interpretation, and faithful model)**.** An interpretation $[\![\, \cdot \,]\!]_G$ of a graphical calculus is called faithful if no two generators are sent to the same morphism. A model of RING-PROP$_S$ in G is faithful if no two generators of RING-PROP$_S$ are sent to the same morphism by the composition of $\mathcal{M}_S$ and $[\![\, \cdot \,]\!]_G$.

Definition 5.18 (Treating RING-PROP$_R$ as a graphical calculus)**.** We create the (explicitly not compact-closed) graphical calculus RING-PROP$_R^{[\![\,]\!]}$ as the PROP RING-PROP$_R$ of Definition 5.1 with a faithful interpretation $[\![\, \cdot \,]\!]$ into K-bit.

Lemma 5.19. *In* RING-PROP$_R^{[\![\,]\!]}$

$$
\forall r \in R \quad \left[\!\!\left[\; \underset{(r)}{\overset{\shortmid}{\shortmid}} \; \right]\!\!\right] \neq 0 \tag{5.37}
$$

Proof. We rely on the fact that for any matrix M, $0 \otimes M = 0$.

$$\text{assume} \quad \left[\!\!\left[\; \underset{r}{\big|} \;\right]\!\!\right] = 0 \tag{5.38}$$

$$\implies \left[\!\!\left[\; \underset{r \;\; 0}{\boxed{\times}} \;\right]\!\!\right] = 0 \qquad\qquad \text{by } 0 \otimes M = 0 \tag{5.39}$$

$$\implies \left[\!\!\left[\; \underset{r \;\; 0}{\boxed{\times}} \;\right]\!\!\right] \overset{=}{\scriptstyle \times} \left[\!\!\left[\; \underset{0}{\big|} \;\right]\!\!\right] = 0 \tag{5.40}$$

$$\implies \left[\!\!\left[\; \underset{0 \;\; 1}{\boxed{+}} \;\right]\!\!\right] = 0 \qquad\qquad \text{by } 0 \otimes M = 0 \tag{5.41}$$

$$\implies \left[\!\!\left[\; \underset{0 \;\; 1}{\boxed{+}} \;\right]\!\!\right] \overset{=}{\scriptstyle +} \left[\!\!\left[\; \underset{1}{\big|} \;\right]\!\!\right] = 0 \tag{5.42}$$

$$\implies \left[\!\!\left[\; \underset{0}{\big|} \;\right]\!\!\right] = \left[\!\!\left[\; \underset{1}{\big|} \;\right]\!\!\right] \tag{5.43}$$

Which contradicts faithfulness of the generators. $\qquad\square$

Lemma 5.20. *In* RING-PROP$_R^{[\![\;]\!]}$

$$\left[\!\!\left[\; \underset{0}{\circ} \;\right]\!\!\right] \neq \left[\!\!\left[\; \underset{1}{\circ} \;\right]\!\!\right] \lambda, \quad \lambda \in K \tag{5.44}$$

Proof. By contradiction

$$\left[\!\!\left[\; \underset{0}{\circ} \;\right]\!\!\right] \overset{=}{\scriptstyle \times} \left[\!\!\left[\; \underset{0 \;\; 0}{\boxed{\times}} \;\right]\!\!\right] \overset{=}{\scriptstyle hyp.} \left[\!\!\left[\; \underset{0 \;\; 1}{\boxed{\times}} \;\right]\!\!\right] \lambda \overset{=}{\scriptstyle \times} \left[\!\!\left[\; \underset{0}{\circ} \;\right]\!\!\right] \lambda \tag{5.45}$$

Note that by Lemma 5.19 we know that $\left[\!\!\left[\; \underset{0}{\circ} \;\right]\!\!\right] \neq 0$, and so because K is a field we know that $\lambda = 1$. Therefore $\left[\!\!\left[\; \underset{0}{\circ} \;\right]\!\!\right] = \left[\!\!\left[\; \underset{1}{\circ} \;\right]\!\!\right]$, which contradicts faithfulness. $\qquad\square$

Lemma 5.21. *The interpretations* $\left[\!\!\left[\; \underset{0}{\circ} \;\right]\!\!\right]$ *and* $\left[\!\!\left[\; \underset{1}{\circ} \;\right]\!\!\right]$ *are*

$$\left[\!\!\left[\; \underset{0}{\big|} \;\right]\!\!\right] := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad\qquad \left[\!\!\left[\; \underset{1}{\big|} \;\right]\!\!\right] := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{5.46}$$

up to change of basis.

Proof. Follows immediately from the non-colinearity shown in Lemma 5.20   □

Remark 5.22. We limit ourselves to interpretations over K-bit rather than R-bits, and so vector spaces rather than modules, because we want to be able to talk about 'up to change of basis'.

Lemma 5.23. *The choice of basis in Lemma 5.21 forces the following interpretations of $\times$ and $+$ in* RING-PROP$_R^{[\![\ ]\!]}$:

$$\left[\!\!\left[\ \boxed{\times}\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \left[\!\!\left[\ \boxed{+}\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{5.47}$$

Proof. The following products form a basis for $K^2 \otimes K^2$

$$\left[\!\!\left[\ \textcircled{0} \otimes \textcircled{0}\ \right]\!\!\right],\ \left[\!\!\left[\ \textcircled{1} \otimes \textcircled{0}\ \right]\!\!\right],\ \left[\!\!\left[\ \textcircled{0} \otimes \textcircled{1}\ \right]\!\!\right],\ \left[\!\!\left[\ \textcircled{1} \otimes \textcircled{1}\ \right]\!\!\right] \tag{5.48}$$

We will use these basis vectors to determine the entries of $\left[\!\!\left[\ \boxed{+}\ \right]\!\!\right]$ and $\left[\!\!\left[\ \boxed{\times}\ \right]\!\!\right]$. We do this by expressing the following equations as diagrams, which should hold under $[\![\ \cdot\ ]\!]$ by soundness of the rules in definition 5.1:

$$\begin{array}{ll} 0 \times 0 = 0 & \qquad 0 \times 1 = 0 \\ 1 \times 0 = 0 & \qquad 1 \times 1 = 1 \\ 0 + 0 = 0 & \qquad 0 + 1 = 1 \\ 1 + 1 = 1 & \end{array} \tag{5.49}$$

We perform the first calculation (which determines the first column in the matrix interpretation of $\times$) in full as an example:

$$\left[\!\!\left[\ \boxed{\times}\ \right]\!\!\right] \circ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \left[\!\!\left[\ \boxed{\times}\!\!\begin{smallmatrix}\textcircled{0}\ \textcircled{0}\end{smallmatrix}\ \right]\!\!\right] \stackrel{=}{\scriptstyle\times} \left[\!\!\left[\ \textcircled{0}\ \right]\!\!\right] = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5.50}$$

In this manner we are able to determine the entries for the matrix interpretation of $\times$ as:

$$\left[\!\!\left[\ \times\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.51}$$

We do not, however, have enough equations involving addition in (5.49) to determine all the entries in the matrix interpretation of $+$. We have merely determined the first three columns, expressed as:

$$\left[\!\!\left[\ +\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & b \\ 0 & 1 & 1 & c \end{pmatrix} \tag{5.52}$$

Where b and c are elements of K. Since R is a ring; even if we don't know its characteristic we can still define $2 := 1 + 1$.

$$\left[\!\!\left[\ \textcircled{2}\ \right]\!\!\right] \overset{=}{{}_{+}} \left[\!\!\left[\ +\ \textcircled{1}\ \textcircled{1}\ \right]\!\!\right] \overset{=}{{}_{(5.52)}} \begin{pmatrix} 1 + b \\ 2 + c \end{pmatrix} \tag{5.53}$$

We then use the equation $2 \times 0 = 0$ to determine b:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left[\!\!\left[\ \boxed{0}\ \right]\!\!\right] \overset{=}{{}_{\times}} \left[\!\!\left[\ \times\ \textcircled{2}\ \textcircled{0}\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 + b \\ 0 \\ 2 + c \\ 0 \end{pmatrix} \tag{5.54}$$

$$\tag{5.55}$$

$$\implies b = 0 \tag{5.56}$$

We can now determine that $c \in \{0, -1, -2\}$ by checking $2 + 2 = 2 \times 2$:

$$\left[\!\!\left[\ +\ \textcircled{2}\ \textcircled{2}\ \right]\!\!\right] \overset{=}{{}_{(5.52)}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & c \end{pmatrix} \circ \begin{pmatrix} 1 \\ 2 + c \\ 2 + c \\ (2 + c)^2 \end{pmatrix} \tag{5.57}$$

$$\overset{=}{{}_{\times}} \left[\!\!\left[\ \times\ \textcircled{2}\ \textcircled{2}\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 2 + c \\ 2 + c \\ (2 + c)^2 \end{pmatrix} \tag{5.58}$$

$$\therefore \quad 2(2 + c) + c(2 + c)^2 = (2 + c)^2 \tag{5.59}$$

$$\therefore \quad (2+c)(c^2 + 2 + c) = 0 \tag{5.60}$$

$$\therefore \quad c \in \{0, -1, -2\} \tag{5.61}$$

Finally we show that $c = 0$ by contradiction:

- If $c = -1$

$$\left[\!\!\left[\; \vcenter{\hbox{\includegraphics{plus}}} \;\right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \left[\!\!\left[\; \vcenter{\hbox{\includegraphics{one}}} \;\right]\!\!\right] \tag{5.62}$$

$$\implies 1 + 1 = 1 \tag{5.63}$$

$$\implies 1 = 0 \quad \lightning \tag{5.64}$$

- If $c = -2$:

$$\left[\!\!\left[\; \vcenter{\hbox{\includegraphics{plus}}} \;\right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left[\!\!\left[\; \vcenter{\hbox{\includegraphics{zero}}} \;\right]\!\!\right] \tag{5.65}$$

$$\implies 1 + 1 = 0 \tag{5.66}$$

$$\tag{5.67}$$

If $1 + 1 = 0$ then either we are working in characteristic 2 (and $c = -2 = 0$) or we have a contradiction.

Therefore $c = 0$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Remark 5.24. We chose this basis so that multiplication would coincide with the usual interpretation of the Z-spider. There is an argument for using the computational basis $\left[\!\!\left[\; \vcenter{\hbox{\includegraphics{zero}}} \;\right]\!\!\right] = |\,0\,\rangle$ and $\left[\!\!\left[\; \vcenter{\hbox{\includegraphics{one}}} \;\right]\!\!\right] = |\,1\,\rangle$ because it makes the actions of $\times$ and $+$ easier to read in bra-ket notation, but it also makes the results of the next section harder to see. Under the computational basis the interpretations of $\times$ and $+$ are:

$$\left[\!\!\left[\; \vcenter{\hbox{\includegraphics{times}}} \;\right]\!\!\right]_{\text{computational}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.68}$$

$$\left[\!\!\left[\; \vcenter{\hbox{\includegraphics{plus}}} \;\right]\!\!\right]_{\text{computational}} = \begin{pmatrix} 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{pmatrix} \tag{5.69}$$

Now that we have established the interpretation of $\text{RING-PROP}_R^{[\![\,]\!]}$ up to a change of basis, we shall look at the relationship this forces between R and K as rings. First we look at the *sets* R and K (Lemma 5.25), and then find an injective ring homomorphism from R to K (Lemma 5.26).

Lemma 5.25. *In* $\text{RING-PROP}_R^{[\![\,]\!]}$, *recalling that we assume the interpretation into K-bit is faithful, R embeds into K as sets.*

Proof. Using the interpretation from lemma 5.21 (i.e. up to change of basis) we know that:

$$r \in R$$

$$\left[\!\!\left[\; \begin{matrix} | \\ r \end{matrix} \; \right]\!\!\right] = \begin{pmatrix} b \\ c \end{pmatrix} \tag{5.70}$$

$$\implies \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left[\!\!\left[\; \begin{matrix} | \\ 0 \end{matrix} \; \right]\!\!\right] \underset{\times}{=} \left[\!\!\left[\; \begin{matrix} \times \\ r \quad 0 \end{matrix} \; \right]\!\!\right] = \begin{pmatrix} b \\ 0 \end{pmatrix} \tag{5.71}$$

$$\implies b = 1 \tag{5.72}$$

Therefore we can construct the function f that sends an element of the set R to its second component in $\left[\!\!\left[\; \begin{matrix} | \\ r \end{matrix} \; \right]\!\!\right]$.

$$f : R \to K \tag{5.73}$$

$$r = \begin{pmatrix} 1 \\ c \end{pmatrix} \mapsto c \tag{5.74}$$

Since the interpretation is faithful, and the first component is always 1, f must be injective. $\qquad\square$

Proposition 5.26. *For* $\text{RING-PROP}_R^{[\![\,]\!]}$ *over K-bit there is an injective ring homomorphism $R \hookrightarrow K$*

Proof. The restriction of K to R in Lemma 5.25 preserves all of $(\times, +, 0, 1)$ so f is a ring homomorphism. $\qquad\square$

Remark 5.27. This, in a sense, classifies phase ring substructures of a graphical calculus over K-bit. That is whenever one finds a model for RING-PROP_R (Definition 5.1) then Proposition 5.26 shows that R must be a subring of K. Note that the calculus RING_K contains a model of RING-PROP_R for every subring R of K, so this subring requirement is the strictest subring requirement we will be able to find.

Now that we have shown how restrictive $\text{RING-PROP}_R^{[\![\,]\!]}$ is in terms of its requirements on R, let us look at how expressive it is as a calculus, and compare it to the earlier parts of this chapter.

Lemma 5.28. $\text{RING-PROP}_R^{[\![\,]\!]}$ *is not universal over K-bit*

Proof. We cannot form morphisms of the shape $1 \to 0$ $\square$

This is unsurprising; the category K-bit is compact closed, but RING-PROP_R is not. What is surprising is that adding compact closure, and setting R to K, *is* enough to achieve universality. Not only this but we will show that our newly added generators again have their interpretations determined by our choice of basis.

Definition 5.29 ($\text{RING-PROP}_R^{T,\,[\![\,]\!]}$)**.** The graphical calculus $\text{RING-PROP}_R^{T,\,[\![\,]\!]}$ is defined by adding the generators

$$\text{cup} \qquad \smile \tag{5.75}$$

$$\text{cap} \qquad \frown \tag{5.76}$$

to $\text{RING-PROP}_R^{[\![\,]\!]}$, requiring that they obey the snake equations

$$\text{(5.77)}$$

and also enact the transpose:

$$\left[\!\!\left[\; \boxed{D} \;\right]\!\!\right] = [\![\, D \,]\!]^T \tag{5.78}$$

Lemma 5.30. *The requirement given in Definition 5.29 forces the following interpretation for the cup and cap in* $\text{RING-PROP}_R^{T,\,[\![\,]\!]}$:

$$\left[\!\!\left[\; \smile \;\right]\!\!\right] = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \qquad\qquad \left[\!\!\left[\; \frown \;\right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \tag{5.79}$$

Proof.

$$
\left[\!\!\left[\ \vcenter{\hbox{a}}\ \right]\!\!\right] = \left[\!\!\left[\ \vcenter{\hbox{$\frown$}}\ \right]\!\!\right] \circ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ a & 0 \\ 0 & a \end{pmatrix} \underset{(5.78)}{=} \begin{pmatrix} 1 & a \end{pmatrix} \tag{5.80}
$$

$$
\Longrightarrow \left[\!\!\left[\ \vcenter{\hbox{$\frown$}}\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \tag{5.81}
$$

We now use the requirement that the cup and cap obey the snake equation alongside (5.81) to find the interpretation for the cup:

$$
\vcenter{\hbox{snake}} = \vcenter{\hbox{$|$}} \qquad\qquad \Longrightarrow \left[\!\!\left[\ \vcenter{\hbox{$\smile$}}\ \right]\!\!\right] = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{5.82}
$$

$\square$

Proposition 5.31. *The graphical calculus* $\mathrm{RING\text{-}PROP}_K^{T,\ [\![\ ]\!]}$ *has the same generators and interpretation as* RING_K, *up to change of basis.*

Proof. This is a combination of Lemmas 5.21, 5.23 and 5.30, and Definitions 4.1, 5.1 and 5.29. $\square$

Proposition 5.32. *For every generic object* $(G, \mathcal{M}_R)$ *of* **Ring-GC** $(K\text{-}bit)$ *there is a unique map* $\psi : K\text{-}bit \to K\text{-}bit$ *dependent only on* $(G, \mathcal{M}_R)$ *such that for any morphism* $f : \mathrm{RING}_K \to (G, \mathcal{M}_R)$ *the following diagram commutes*

$$
\begin{array}{ccc}
\mathrm{RING}_K & \xrightarrow{\ f\ } & (G, \mathcal{M}_R) \\
{\scriptstyle [\![\cdot]\!]_{\mathrm{RING}_K}}\Big\downarrow & & \Big\downarrow{\scriptstyle [\![\cdot]\!]_G} \\
K\text{-}bit & \xrightarrow[\ \psi\ ]{} & K\text{-}bit
\end{array} \tag{5.83}
$$

What's more ψ_G *is given by a change of basis.*

Proof. We construct ψ as simply the change of basis that sends

$$
\left[\!\!\left[\ \vcenter{\hbox{0}}\ \right]\!\!\right]_{\mathrm{RING}_K} \mapsto \left[\!\!\left[\ g_0\ \right]\!\!\right]_G \qquad\qquad \left[\!\!\left[\ \vcenter{\hbox{1}}\ \right]\!\!\right]_{\mathrm{RING}_K} \mapsto \left[\!\!\left[\ g_1\ \right]\!\!\right]_G \tag{5.84}
$$

which by Lemma 5.20 is indeed a change of basis (because these states form a basis), and by Lemma 5.23 the diagram commutes for all generators of RING_K. $\square$

It is easy to construct trivial, or at least dull, graphical calculi in **Ring-GC** (K). For example for $R \subsetneq K$ the calculus RING_R is an object of **Ring-GC** (K) (see Definition 5.8). What we would like to show is a relationship between RING_K and any 'suitably expressive' object of **Ring-GC** (K). It turns out that completeness and universality of the graphical calculus is suitable for our needs.

Lemma 5.33. *Any object $(G, \mathcal{M}_R)$ of* **Ring-GC** (K) *that is a complete universal graphical calculus (over K-bit) extends naturally to an object $(G, \mathcal{M}_K)$.*

Proof. Consider the object $(G, \mathcal{M}_R)$. Let $f : R \to K$ be the f from Proposition 5.26 that identifies R with a subring of K. For every $k \in K \setminus f(R)$ there is a representative D in G of the sole equivalence class with

$$[\![\, D \,]\!]_G = \begin{pmatrix} 1 \\ k \end{pmatrix} \tag{5.85}$$

(with respect to the basis of Lemma 5.21) because the calculus G is universal and complete.

Choose, for each $k \in K \setminus f(R)$ such a diagram and label it g_k. In doing so we have constructed $(G, \mathcal{M}_K)$; an object in **Ring-GC** (K) with the same generators as the original object but with phase ring K. The multiplication and addition rules of RING-PROP_K hold in G because the semantics of g_+ and $g_\times$ have already been determined by Lemma 5.23, those semantics preserve addition and multiplication with g_k, and G is complete. $\qquad\square$

Note that we needed to be able to determine the interpretation (up to change of basis) in order to prove Lemma 5.33. Since we can extend complete, universal graphical calculi over K-bit naturally to ones with models $\mathcal{M}_K$, let us look at the category of just those graphical calculi with model $\mathcal{M}_K$, and where the morphisms preserve this 'fullness' of the model.

Definition 5.34 (K-Phase-Field Calculi)**.** We define the category of K-Phase-Field Graphical Calculi as the subcategory of **Ring-GC** (K) where the objects G_S are restricted by requiring $S \cong K$, and the ring homomorphisms $f_\mathbb{R}$ are restricted to being automorphisms of K.

Proposition 5.35. *The object RING_K is 'initial, up to ring isomorphism', for the category of K-Phase Field Graphical Calculi. I.e. every morphism f from RING_K to G_K is given by a phase ring isomorphism $\hat{\phi} : \text{RING}_K \to \text{RING}_K$ (see Definition 5.16) followed by an unique morphism $(f_{\mathbb{D},2}, f_{\mathbb{R},2}) : \text{RING}_K \to G$ which is the identity on the ring structure:*

$$f_{\mathbb{D},2} : \text{RING}_K \to G \tag{5.86}$$

$$\underset{k}{|} \mapsto g_k \qquad\qquad \boxed{+} \mapsto g_+ \qquad\qquad \boxed{\times} \mapsto g_\times \tag{5.87}$$

$$f_{\mathbb{R},2} = id_K : K \to K \tag{5.88}$$

What's more, every morphism of this form exists.

Proof. We first show that every morphism of the form $(f_{\mathbb{D},2}, f_{\mathbb{R},2}) : \text{RING}_K \to G$, as constructed in the statement, obeys the requirements of our category. Given a field automorphism $\phi : K \to K$:

$$f_{\mathbb{D},2}\phi_{\mathbb{D}}\left(\begin{array}{c} | \\ \boxed{s} \end{array}\right) = f_{\mathbb{D},2}\left(\begin{array}{c} | \\ \boxed{\phi(s)} \end{array}\right) = g_{id_K\phi(s)} = g_{\phi(s)} \tag{5.89}$$

$$f_{\mathbb{D},2}\phi_{\mathbb{D}}\left(\begin{array}{c} | \\ \boxed{+} \\ \diagup\diagdown \end{array}\right) = f_{\mathbb{D},2}\left(\begin{array}{c} | \\ \boxed{+} \\ \diagup\diagdown \end{array}\right) = g_+ \tag{5.90}$$

$$f_{\mathbb{D},2}\phi_{\mathbb{D}}\left(\begin{array}{c} | \\ \boxed{\times} \\ \diagup\diagdown \end{array}\right) = f_{\mathbb{D},2}\left(\begin{array}{c} | \\ \boxed{\times} \\ \diagup\diagdown \end{array}\right) = g_\times \tag{5.91}$$

The morphism $(f_{\mathbb{D},2}, f_{\mathbb{R},2})$ is unique because $f_{\mathbb{D},2}$ is entirely determined by the action on the generators of RING_K, which is entirely determined by the model $\mathcal{M}_K$ in G_K. □

Remark 5.36. We have therefore shown that:

- RING_K contains only those generators needed to be a phase ring graphical calculus over monoids (Definition 4.1 and Proposition 5.31)

- RING_K is universal and complete (Theorem 4.3)

- RING_K has its interpretation fully determined, up to choice of basis (Lemma 5.23)

- RING_K is 'initial, up to field automorphism' for K-Phase-Field Graphical Calculi (Proposition 5.35)

- Every complete universal graphical calculus in **Ring-GC** (K) extends to a K-Phase-Field Graphical Calculus (Lemma 5.33)

Which, in the view of the author, is sufficient reason for calling RING_K *the* generic phase field calculus for the field K into K-bits.

It is tempting to quotient out this 'up to isomorphism' aspect of Proposition 5.35, but doing so would remove phase ring homomorphisms. These special morphisms will turn out to have important properties, in particular that they send sound equations to sound equations, i.e. 'theorems are closed under phase ring homomorphism'.

5.3 Phase ring homomorphisms

In Definition 5.16 we showed that any ring homomorphism $\phi : S \to S'$ lifts to the maps

$$\hat{\phi} : \text{RING}_S \to \text{RING}_{S'} \tag{5.92}$$

$$\tilde{\phi} : S\text{-bit} \to S'\text{-bit} \tag{5.93}$$

In this section we continue this idea, showing first how to perform the same operation for ZW_R and ZH_R, and then how this lifting interacts with the semantics (Theorem 5.42) and syntax (Theorem 5.46) of all three calculi. We will assume, when speaking of ZH_R, that the ring R has $\frac{1}{2}$. Exploration of algebras other than rings is delayed until §5.4.

Proposition 5.37. *For $\hat{\phi}$ and $\tilde{\phi}$ from Definition 5.16 the following diagram commutes*

$$\begin{array}{ccc} \text{RING}_S & \xrightarrow{\hat{\phi}} & \text{RING}_{S'} \\ \downarrow{\llbracket \cdot \rrbracket} & & \downarrow{\llbracket \cdot \rrbracket} \\ S\text{-bit} & \xrightarrow{\tilde{\phi}} & S'\text{-bit} \end{array} \tag{5.94}$$

Proof. We verify that the diagram (5.94) commutes for all the generators of RING_S:

$$\tag{5.95}$$

$$\tag{5.96}$$

$$\begin{array}{ccc} & \xrightarrow{\;\hat{\phi}\;} & \\ & & \\ \Big\downarrow \llbracket \,\cdot\, \rrbracket_S & & \Big\downarrow \llbracket \,\cdot\, \rrbracket_{S'} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \xrightarrow{\;\tilde{\phi}\;} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \tag{5.97}$$

$\square$

Proposition 5.38. *The analogous lifting of ϕ to $\hat{\phi}$ for ZW_R and ZH_R are as follows:*

$$\phi : S \to S' \tag{5.98}$$

$$\hat{\phi} : ZW_S \to ZW_{S'} \tag{5.99}$$

 (5.100)

 (5.101)

$$\hat{\phi} : ZH_S \to ZH_{S'} \tag{5.102}$$

 (5.103)

 (5.104)

And the following diagrams commute

$$\begin{array}{ccc} ZW_S & \xrightarrow{\;\hat{\phi}\;} & ZW_{S'} \\ \Big\downarrow \llbracket \,\cdot\, \rrbracket & & \Big\downarrow \llbracket \,\cdot\, \rrbracket \\ S\text{-}bit & \xrightarrow{\;\tilde{\phi}\;} & S'\text{-}bit \end{array} \qquad \begin{array}{ccc} ZH_S & \xrightarrow{\;\hat{\phi}\;} & ZH_{S'} \\ \Big\downarrow \llbracket \,\cdot\, \rrbracket & & \Big\downarrow \llbracket \,\cdot\, \rrbracket \\ S\text{-}bit & \xrightarrow{\;\tilde{\phi}\;} & S'\text{-}bit \end{array} \tag{5.105}$$

Note that we do not make it notationally explicit which base language is being considered. The construction of $\hat{\phi}$ and $\tilde{\phi}$ are the same for RING_R, ZH, and ZW, with the context making clear which language we are considering.

Proof. In both cases $\tilde{\phi}$ is defined analogously to Definition 5.16, and one can verify that for the generators of ZH and ZW that in both cases we have $\llbracket \, \hat{\phi} D \, \rrbracket = \tilde{\phi} \llbracket \, D \, \rrbracket$. It is also clear from the definitions that $(\hat{\phi}, \phi)$ satisfies the conditions for being a morphism

in **Ring-GC** (R-bit). $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Remark 5.39. Since RING, ZH and ZW are all universal graphical calculi we note that $\tilde{\phi}$ is entirely determined by its action on the generators.

Since ϕ determines $\hat{\phi}$ we propose the name 'phase homomorphism pair' for this structure of $\phi, \hat{\phi}, \tilde{\phi}$ and in §5.4 we will give a broader definition of such a pair. §5.4 also gives explicit examples in the phase *group* case. The primary property of these pairs is preservation of *soundness*:

Proposition 5.40. *If the diagrammatic equation $A = B$ in RING_S is sound, then the diagrammatic equation $\hat{\phi}(A) = \hat{\phi}(B)$ in $\mathrm{RING}_{S'}$ is sound*

Proof.

$$[\![\, A \,]\!] = [\![\, B \,]\!] \implies \tilde{\phi}[\![\, A \,]\!] = \tilde{\phi}[\![\, B \,]\!] \xRightarrow{5.37} [\![\, \hat{\phi}A \,]\!] = [\![\, \hat{\phi}B \,]\!] \qquad (5.106)$$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Example 5.41 (Soundness of field phase homomorphisms). Consider the field automorphisms of $\mathbb{Q}[i, \sqrt{2}]$

$$\sigma : a + ib + \sqrt{2}c + i\sqrt{2}d \mapsto a - ib + \sqrt{2}c - i\sqrt{2}d \qquad (5.107)$$
$$\tau : a + ib + \sqrt{2}c + i\sqrt{2}d \mapsto a + ib - \sqrt{2}c - i\sqrt{2}d \qquad (5.108)$$

The following $\mathrm{RING}_{\mathbb{Q}[i,\sqrt{2}]}$ equation is sound:

 $\qquad (5.109)$

Therefore the following $\mathrm{RING}_{\mathbb{Q}[i,\sqrt{2}]}$ equations, found by applying the automorphisms $\hat{\sigma}$, $\hat{\tau}$ and $\hat{\sigma}\hat{\tau}$, are sound:

$\hat{\sigma}$ $\qquad\qquad\qquad$ $\qquad (5.110)$

$$\hat{\tau} \qquad \qquad \tag{5.111}$$

$$\hat{\sigma}\hat{\tau} \qquad \qquad \tag{5.112}$$

$$\tag{5.113}$$

Theorem 5.42 (Phase homomorphisms preserve soundness). *if $\phi : R \to S$ is a ring homomorphism then $\mathrm{RING}_R \vDash A = B$ implies $\mathrm{RING}_S \vDash \hat{\phi}A = \hat{\phi}B$. The same is true for ZW_R and ZH_R.*

Proof. The statement for RING is just Proposition 5.40. For ZW and ZH use the same argument as Proposition 5.40 in conjunction with Proposition 5.3.   □

We can even go beyond the level of semantics, and show that syntactic entailment also commutes with $\hat{\phi}$ (for these three calculi). We will need to show two things: That rules are preserved under the action of $\hat{\phi}$, and that rule application is preserved under the action of $\hat{\phi}$. Preservation of the rules is done by checking all the rules for every calculus mentioned by hand. This requires no additional calculation, and so we include just one example below. There are rules in ZX that do *not* satisfy the analogous requirements for phase group homomorphisms, which we will mention explicitly in the proof of Proposition 5.54.

Example 5.43 (A rule preserved by $\hat{\phi}$). Given a ring homomorphism $\phi : R \to S$, the spider rule of RING_R is mapped by $\hat{\phi}$ to a restriction of the same rule in RING_S:

$$\tag{5.114}$$

The restriction is that the above equation only matches onto phases that are in the image of ϕ in S.

Proposition 5.44. *Using the rules from the completeness theorem for RING. (Theorem 4.3) and given a ring homomorphism $\phi : R \to S$, if $\mathrm{RING}_R \vdash A = B$ then $\mathrm{RING}_S \vdash \hat{\phi}A = \hat{\phi}B$, using the same proof steps.*

Proof. We first note that for any rule $A = B$ in RING_R, the equation $\hat{\phi}A = \hat{\phi}B$ expresses a restriction of the rule $A = B$ in RING_S to the image of R under ϕ. This is because

every phase used in a rule in RING_R is either an integer or a free variable, and so under the image of ϕ is either fixed (for an integer) or expressed as the image of a free variable (for a free variable). For one example see Example 5.43, or for another: Applying $\hat{\phi}$ to the rule (L) (Figure 4.4) in $\mathrm{RING}_\mathbb{Z}$ yields the identical rule (L) in $\mathrm{RING}_\mathbb{Q}$, because tr_z only involves the phases 0 and 1 (recall that we leave the white spider blank when it has phase 1), which are fixed by $\hat{\phi}$.

The following diagram[1], where both the front and back wide rectangles are double-pushout rewrites, commutes, and $\hat{\phi}A = \hat{\phi}B$ is a valid rule application in RING_R:
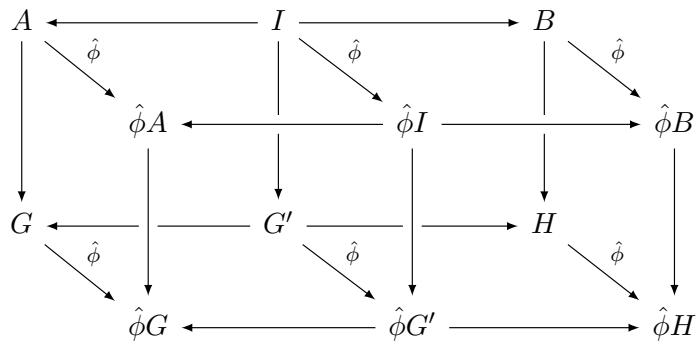
In particular note that

$$
\begin{array}{ccc}
\hat{\phi}A \longleftarrow \hat{\phi}I & & \hat{\phi}I \longrightarrow \hat{\phi}B \\
\downarrow \qquad \downarrow & \text{and} & \downarrow \qquad \downarrow \\
\hat{\phi}G \longleftarrow \hat{\phi}G' & & \hat{\phi}G' \longrightarrow \hat{\phi}H
\end{array}
\tag{5.115}
$$

are both still pushouts, since $\hat{\phi}$ changes none of the underlying graph and only the labels. Therefore a sequence of DPO rewrites from C to D in RING_R becomes a sequence of DPO rewrites from $\hat{\phi}C$ to $\hat{\phi}D$ in RING_S under the action of $\hat{\phi}$.   □

Example 5.45 (Proof preservation of field phase homomorphisms). Using the same equation as in Example 5.41, we see that the proof

$$
\mathrm{RING}_\mathbb{C} \vdash A = B
\tag{5.116}
$$

$$\tag{5.117}$$

---
[1] A *double double pushout diagram.*

is translated, using the field automorphism $\sigma : \mathbb{C} \to \mathbb{C}$, $\sigma : i \mapsto -i$, to the proof

$$\text{RING}_{\mathbb{C}} \vdash \hat{\sigma} A = \hat{\sigma} B \tag{5.118}$$

$$\tag{5.119}$$

Theorem 5.46 (Phase homomorphisms preserve proofs)**.** *If* $\text{RING}_R \vdash A = B$, *then* $\text{RING}_S \vdash \hat{\phi}(A) = \hat{\phi}(B)$ *for every ring homomorphism* $\phi : R \to S$. *The proof of* $\text{RING}_S \vdash \hat{\phi}(A) = \hat{\phi}(B)$ *is constructed as the one-to-one translation via* $\hat{\phi}$ *of the proof of* $\text{RING}_R \vdash A = B$. *The same is true for* ZW_R *and* ZH_R.

Proof. The claim about RING_R follows from Proposition 5.44. One then goes through each rule of ZH_R and ZW_R and shows that the action of $\hat{\phi}$ on any instance of that rule is to create an instance of a rule already present in ZW_S or ZH_S accordingly. We give the (A) rule in ZH_R as an example:

$$\phi : R \to S \qquad \frac{1}{2} \in R \tag{5.120}$$

$$1 \mapsto 1 \tag{5.121}$$

$$2 \mapsto 2 \tag{5.122}$$

$$\therefore \phi : \frac{1}{2} \mapsto \frac{1}{2} \tag{5.123}$$

$$\therefore \hat{\phi} : \quad \mapsto \quad \tag{5.124}$$

$$\tag{5.125}$$

Any assignment of elements of R to a or b in the rule (A) gives a valid instance of the rule $\hat{\phi}(A)$, because $\phi(a) \in S$, $\phi(b) \in S$, $\phi(2) = 2$ and $\phi\left(\frac{a+b}{2}\right) = \frac{\phi(a)+\phi(b)}{2}$. $\qquad \square$

Phase ring homomorphisms shall reappear in the chapter on Conjecture Inference (§7), where we note that these lifted homomorphisms correspond to certain symmetries in a geometric space. They then come up again in the Conjecture Verification chapter (§8) where we use Theorem 5.42 to 'generate' new sound equations from old, and in doing so construct sufficient evidence to then prove hypotheses. Before moving on to conjecture synthesis we shall talk about algebras other than rings.

5.4   Phase algebra homomorphisms and beyond

While ZX is *not* a phase ring graphical calculus we can still consider phase group homomorphisms, just as we did for phase ring homomorphisms earlier. In this section we will classify the phase group homomorphism pairs for the finite subgroups of $[0, 2\pi)$ that contain $\pi/4$. Note that completeness of these finite subgroups has been shown in Ref. [JPV19, Figure 4]. We shall be considering the presentation of ZX where the generators are green spiders with phases from our chosen fragment, and Hadamard gates.

$$\left\{ \raisebox{-1em}{\includegraphics{placeholder}} \ , \ \raisebox{-1em}{\includegraphics{placeholder}} \right\}_{\alpha \in G} \tag{5.126}$$

Definition 5.47 (ZX phase homomorphism)**.** For a given group homomorphism $\phi : A \to B$ we can define the ZX_A phase homomorphism $\hat{\phi}$ as

$$\hat{\phi} : \quad \raisebox{-1em}{\alpha} \quad \mapsto \quad \raisebox{-1em}{\phi(\alpha)} \tag{5.127}$$

$$\raisebox{-1em}{\includegraphics{placeholder}} \quad \mapsto \quad \raisebox{-1em}{\includegraphics{placeholder}} \tag{5.128}$$

Remark 5.48. Something similar to phase group homomorphisms has already been used in the ZX calculus. Some of the early incompleteness proofs used 'non-standard interpretations', such as in Refs [DP09, Lemma 8], [DP14, Lemma 1.5] or [SZ14, §2]. These papers do not define something akin to our $\hat{\phi}$, but instead jump straight to defining and using $\left[\!\left[\hat{\phi}(\cdot) \right]\!\right]$. The idea of $\hat{\phi}$ is already apparent in the definition of $\left[\!\left[\cdot \right]\!\right]_k$ of Ref. [SZ14].

At the end of the chapter we will have the language for defining the general case of a phase homomorphism and phase homomorphism pairs (Definition 5.62) but for now we define a ZX phase homomorphism pair as:

Definition 5.49 (ZX phase homomorphism pair)**.** A ZX phase homomorphism pair $(\phi, \tilde{\phi})$ is a group homomorphism $\phi : A \to B$ and a functor $\tilde{\phi} : R\text{-bit} \to S\text{-bit}$ such that the following diagram commutes:

$$\begin{array}{ccc} A & ZX_A \xrightarrow{\ \left[\!\left[\cdot \right]\!\right]\ } R\text{-bit} \\ \downarrow{\phi} & \downarrow{\hat{\phi}} \qquad\qquad \downarrow{\tilde{\phi}} \\ B & ZX_B \xrightarrow{\ \left[\!\left[\cdot \right]\!\right]\ } S\text{-bit} \end{array} \tag{5.129}$$

Proposition 5.50. *The finite subgroups of $[0, 2\pi)$ under addition are cyclic groups of*

the form $< 2\pi i k/n >$, and have endomorphisms of the form

$$\phi_j : 2\pi i k/n \mapsto j \times 2\pi i k/n \tag{5.130}$$

These endomorphisms lift to a phase homomorphism pair (Definition 5.49) if and only if the map $\tilde{\phi}$

$$\tilde{\phi} : \mathbb{Z}[2^{-\frac{1}{2}}, e^{2\pi i k/n}]\text{-}bit \to \mathbb{Z}[2^{-\frac{1}{2}}, e^{2\pi i k/n}]\text{-}bit \tag{5.131}$$

$$e^{2\pi i k/n} \mapsto e^{2\pi i j k/n} \tag{5.132}$$

is a well-defined ring homomorphism (when restricted to scalars) that fixes the subring $\mathbb{Z}[2^{-\frac{1}{2}}]$.

Proof. The statements on the presentation of the finite subgroups and their endomorphisms follow from clearing denominators in the generators. The statement about lifting to phase homomorphisms is shown by noting that:

- $\tilde{\phi}$ must be a ring homomorphism when restricted to scalars

- For the green spider the diagram in (5.129) commutes for any ϕ that fixes 1 and 0, but for the Hadamard node we additionally require $\tilde{\phi}$ to fix -1 and $\frac{1}{\sqrt{2}}$

$\square$

Example 5.51 (Clifford+T ZX phase homomorphism pairs). Consider the Clifford+T phase group $< e^{2\pi i/8} >$ and its relationship to the field automorphisms σ and τ from Example 5.41

$$\sigma : i \mapsto -i \tag{5.133}$$

$$\tau : \sqrt{2} \mapsto -\sqrt{2} \tag{5.134}$$

Note that τ does not fix $\sqrt{2}$. In Figure 5.1 we show all the endomorphisms of the Clifford+T phase group, the associated ring homomorphism $\tilde{\phi}$ from (5.132), and whether $(\phi, \tilde{\phi})$ forms a ZX phase homomorphism pair. If no ring homomorphism is recorded in the column then the $\tilde{\phi}$ constructed is not a ring homomorphism at all. If $\tilde{\phi}$ is a valid ring homomorphism and also fixes $\sqrt{2}$, then $(\phi, \tilde{\phi})$ forms a phase homomorphism pair, and we record the effect of the pair in the final column. Note that this table shows that the only phase homomorphism pairs of the Clifford+T ZX phase group are the identity and complex conjugation (i.e. ϕ_1 and ϕ_7).

Proposition 5.52. *For a finite fragment of ZX that contains $\pi/4$, a group endomorphism ϕ_j lifts to a phase homomorphism pair if and only if $j \equiv 1$ or 7 modulo 8.*

Proof. Restrict ϕ_j to the subgroup $< \pi/4 >$, and note that by Figure 5.1 j must be 1 or 7 modulo 8. Conversely if $j \equiv 1$ then ϕ_j fixes $\sqrt{2}$ and i, and if $j \equiv 7$ then ϕ_j moves

Group endomorphism	$\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ automorphism	Pair
ϕ_0	–	–
ϕ_1	Identity	Identity
ϕ_2	–	–
ϕ_3	τ	–
ϕ_4	–	–
ϕ_5	$\sigma\tau$	–
ϕ_6	–	–
ϕ_7	σ	Complex conjugation

Figure 5.1: The group endomorphisms of the Clifford+T phases, details on whether they lift to phase group homomorphisms, and the associated ring automorphism of $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$

i but fixes $\sqrt{2}$ (as seen by the action on $\frac{1+i}{\sqrt{2}}$), in both cases satisfying the conditions of 5.50. $\qquad\square$

Remark 5.53. We have already classified the phase ring homomorphisms for ZW, ZH and RING by showing that every ring homomorphism lifts to a phase ring homomorphism pair.

Looking now at the phase group version of Theorem 5.46 (phase ring homomorphisms preserve proofs) we need a set of rules for these fragments of ZX, for which we take the rules of Figure 1 of Ref. [JPV19], along with the rule (cancel) given in their Definition 10. We call this set of rules 'RZX'.

Proposition 5.54. *For a finite fragment of ZX that contains $\pi/4$, a group endomorphism ϕ_j lifts to a RZX-proof preserving map if and only if $j \equiv 1 \mod 8$*

$$RZX \vdash A = B \quad \Longrightarrow \quad RZX \vdash \hat{\phi}A = \hat{\phi}B \tag{5.135}$$

Proof. Following the same structure as the proof of Theorem 5.46 we note that the rules of RZX contain phases that are multiples of $\pi/4$, and in particular any ϕ_j with $j \equiv 7 \mod 8$ will send the rule (BW) (shown in Figure 5.2) to an equation that is not a rule. In contrast if $j \equiv 1 \mod 8$ then the rules are preserved by $\hat{\phi}$. By Proposition 5.52 the only suitable ϕ_j are those where $j \equiv 1 \mod 8$. $\qquad\square$

Remark 5.55. While the classification of ZX_G phase homomorphism pairs in Proposition 5.52 only applied to finite subgroups of $[0, 2\pi)$ containing $\pi/4$, it is worth remembering that any given ZX diagram only contains finitely many phases, and so this result applies in all situations where the phases are rational multiples of π.

Remark 5.56. The result of Proposition 5.54 does not imply that RZX is in any way deficient. RZX is, after all, still sound, universal, and complete. Rather this result highlights
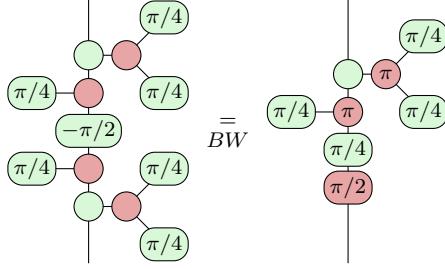
Figure 5.2: The rule (BW) from Ref. [JPV19]

that the notation of ZX does not match perfectly onto the available algebraic structure. There is a sense in which phase ring homomorphisms 'just work' for ZW, RING, and ZH, but the same does not apply to phase group homomorphisms for ZX.

Remark 5.57. It is possible to generalise the above work from rings and groups to Σ-algebras. Σ-algebras are at the heart of Universal Algebra, allowing us to represent algebraic concepts without restricting ourselves to a particular algebraic system. Part of our reason for not doing so from the start lies in the next chapter: ZQ is based on group structure but cannot be represented as a model of the Σ-algebra for groups (Corollary 6.11).

We present the general method of turning a signature and equational theory into a parameterised family of PROPs below, as well as the general definition of a phase homomorphism pair for Σ-algebras. As just mentioned this framework is not sufficient for all our needs, which is why it is covered in such brief terms. It is, however, a bridging between the concepts of universal algebras and graphical calculi.

Definition 5.58 (Signatures, terms, algebras, homomorphisms [BN98, Section 3]).   •
A signature Σ is a set of function symbols with fixed arities. We write $f \in \Sigma^{(n)}$ if f is a function symbol with arity n.

- A Σ-term is inductively defined using a set of variables: Every variable is a term, and for every function symbol $f \in \Sigma^{(n)}$ and terms $t_1, \ldots, t_n$ the application $f(t_1, \ldots, t_n)$ is a term.

- A Σ-algebra $\mathcal{A}$ is a set A, called the carrier set, and an assignment of each function variable $f \in \Sigma^{(n)}$ to a function $f^{\mathcal{A}} : A^n \to A$.

- A Σ-homomorphism $\phi : \mathcal{A} \to \mathcal{B}$ is a function on the carrier sets $A \to B$ such that

$$\phi(f^{\mathcal{A}}(x_1, \ldots, x_n)) = f^{\mathcal{B}}(\phi(x_1), \ldots \phi(x_n)) \tag{5.136}$$

With these definitions we can form a PROP analogue of the same concept. This is

very similar to the idea of a *term tree* [BN98, Figure 3.1] except that the PROP structure is also present, allowing arbitrary horizontal products and wire permutations.

Definition 5.59 (PROP form of a Σ-algebra)**.** Given a signature Σ we create the PROP $\Sigma\mathbb{P}$ generated by the morphisms

$$
\boxed{f} \qquad : n \to 1 \quad \forall f \in \Sigma^{(n)} \tag{5.137}
$$

For any Σ-algebra $\mathcal{A}$ with carrier set A and function assignments $f^{\mathcal{A}}$ we generate the PROP $\Sigma\mathbb{P}_{\mathcal{A}}$ by also including the diagrams

$$
\stackrel{}{\underset{a}{\bigcirc}} \qquad \forall a \in A \tag{5.138}
$$

Given a set of variables X (symbols distinct from A and Σ) we likewise represent them as

$$
\stackrel{}{\underset{x}{\bigcirc}} \qquad \forall x \in X \tag{5.139}
$$

For each function symbol f we have the rewrite rule app_f which applies the function $f^{\mathcal{A}}$ to its inputs:

$$
\boxed{f}\big(x_1 \ \ldots \ x_n\big) \quad \overset{=}{app_f} \quad \big(f^{\mathcal{A}}(x_1,\ldots,x_n)\big) \tag{5.140}
$$

Any Σ-homomorphism $\phi : \mathcal{A} \to \mathcal{A}'$ lifts to a morphism $\bar{\phi} : \Sigma\mathbb{P}_A \to \Sigma\mathbb{P}_B$ of Σ-PROPs which is the identity on function symbols and sends

$$
\stackrel{}{\underset{a}{\bigcirc}} \quad \mapsto \quad \stackrel{}{\underset{\phi(a)}{\bigcirc}} \tag{5.141}
$$

Note that any equational theory one wishes to apply to a signature can be applied *inside* the phases; i.e. if $E \vdash a = b$ then we would consider the following to be an isomorphism of diagrams:

$$
E \vdash a = b \qquad\qquad \therefore \quad \stackrel{}{\underset{a}{\bigcirc}} \cong \stackrel{}{\underset{b}{\bigcirc}} \tag{5.142}
$$

Definition 5.1, where we defined RING-PROP, is an example of such a Σ-PROP. As we did for RING-PROP we can define a *model* of $\Sigma\mathbb{P}_{\mathcal{A}}$ as a strict symmetric monoidal functor $\mathcal{M}_{\mathcal{A}}$ out of $\Sigma\mathbb{P}_{\mathcal{A}}$, and form the following category:

Definition 5.60 (Σ Graphical Calculi)**.** The category of Σ Graphical Calculi over the category $\mathcal{C}$, written $\Sigma\text{-}\mathbf{GC}\,(\mathcal{C})$, is the category given by:

- Objects are pairs $(G, \mathcal{M}_\mathcal{A})$ where G is a (compact closed) graphical calculus given as a collection of generators, an interpretation $[\![\cdot]\!]_G : G \to \mathcal{C}$, and rewrite rules, and $\mathcal{M}_\mathcal{A}$ is a model of $\Sigma\mathbb{P}_\mathcal{A}$

- Morphisms

$$f : (G, \mathcal{M}_\mathcal{A}) \to (G', \mathcal{M}_{\mathcal{A}'}) \tag{5.143}$$

are given by a strict symmetric monoidal functor $f_\mathbb{D} : G \to G'$ as well as a Σ-homomorphism $\phi : \mathcal{A} \to \mathcal{A}'$ such that the right hand diagram commutes:

$$
\begin{array}{ccccc}
\mathcal{A} & & \Sigma\mathbb{P}_\mathcal{A} & \xrightarrow{\mathcal{M}_\mathcal{A}} & G \\
\downarrow{\scriptstyle \phi} & & \downarrow{\scriptstyle \bar{\phi}} & & \downarrow{\scriptstyle f_\mathbb{D}} \\
\mathcal{A}' & & \Sigma\mathbb{P}_{\mathcal{A}'} & \xrightarrow{\mathcal{M}_{\mathcal{A}'}} & G'
\end{array}
\tag{5.144}
$$

The category **Ring-GC** $(R\text{-bit})$ given in Definition 5.5 is an example of a category of Σ Graphical Calculi, where Σ is the usual signature for rings. Our final definitions of this chapter are the general definitions for phase homomorphisms and phase homomorphism pairs. This extends the notion first seen in Definition 5.16 of finding an algebra homomorphism ϕ that lifts to a map $\hat{\phi}$ 'acting on phases' for diagrams and also commutes with the interpretation via some third map $\tilde{\phi}$.

Definition 5.61 (Phase homomorphism)**.** The phase homomorphism for a Σ-homomorphism $\phi : \mathcal{A} \to \mathcal{A}'$ and objects $G_\mathcal{A}$ and $G'_{\mathcal{A}'}$ is the Σ-**GC** $(\mathcal{C})$ morphism $G_\mathcal{A} \to G'_{\mathcal{A}'}$ (if it exists) given by ϕ and $f_\mathbb{D}$ such that:

- $f_\mathbb{D}$ acts as the identity on those vertices not labelled by symbols from Σ or A

- Otherwise the action of $f_\mathbb{D}$ on the label is:

$$a \mapsto \phi(a) \qquad\qquad a \in A \tag{5.145}$$
$$f \mapsto f \qquad\qquad f \in \Sigma^{(n)} \tag{5.146}$$

We refer to such a $f_\mathbb{D}$ as $\hat{\phi}$.

Definition 5.62 (Phase Homomorphism Pair)**.** A phase Σ-homomorphism pair between $G_\mathcal{A}$ and $G'_{\mathcal{A}'}$ in Σ-**GC** $(\mathcal{C})$ is a pair of maps $(\phi, \tilde{\phi})$ such that:

- $\phi : \mathcal{A} \to \mathcal{A}'$ is a Σ-homomorphism

- The phase homomorphism $\hat{\phi} : G \to G'$ exists

- $\tilde{\phi} : \mathcal{C} \to \mathcal{C}$ is a strict symmetric monoidal functor

- The following diagram commutes:

$$
\begin{array}{ccc}
\mathcal{A} & G \xrightarrow{\ \llbracket \,\cdot\, \rrbracket_G\ } \mathcal{C} \\
\downarrow{\scriptstyle\phi} & \downarrow{\scriptstyle\hat{\phi}} \qquad \downarrow{\scriptstyle\tilde{\phi}} \\
\mathcal{A}' & G' \xrightarrow{\ \llbracket \,\cdot\, \rrbracket_{G'}\ } \mathcal{C}
\end{array}
\qquad (5.147)
$$

The commutative square of Definition 5.62 gives us the soundness-preserving property used in Theorem 5.42.

5.5   Summary

The most important result from this chapter (from the point of view of the later chapters) will be that of phase ring homomorphisms preserving soundness (Theorem 5.42), which for the ring calculi we have considered can be paraphrased as 'theorems are closed under phase homomorphism'. These pairs only arise because we view phase algebras as algebras and not sets. We shall rephrase these pairs as symmetries of a geometric space in §7.

We also showed that RING$_\mathbb{C}$ acts as a form of unifier for phase ring qubit graphical calculi. While different choices of generators will yield graphical calculi suited for different situations it is also important to know that any phase ring qubit graphical calculus will have a relationship to RING$_\mathbb{C}$ as described in Remark 5.36.

In the last subsection we also provided a classification of phase group homomorphism pairs for the finite fragments of ZX containing $\pi/4$. Future work would include extending this classification of phase homomorphism pairs to other fragments and calculi, but also the investigation of any other category of Σ Graphical Calculi.

Chapter 6

The Graphical Calculus ZQ

In this chapter:

    ▷ We introduce the graphical calculus ZQ

    ▷ We justify the structural difference between ZQ and ZX

    ▷ We prove completeness of ZQ

    ▷ This chapter touches on the ideas of §5, but stands alone, and is not required for the understanding of later chapters.

The ZX calculus is built from the Z and X classical structures of quantum computing [CD08]. Even in that earliest paper the Z 'phase shift' is illustrated as a rotation of the Bloch Sphere [CD08, §4]. By the time of [Bac16], eight years later, language has changed to that of Z 'rotations' or 'angles' [Bac16, Lemma 3.1.7], and explicit use is made of the Euler Angle Decomposition result; that any rotation in SO3($\mathbb{R}$) can be broken down into rotations about the Z then X then Z axes. The idea behind ZQ is to represent not just the Z and X rotations of the Bloch Sphere, but represent arbitrary rotations via unit-length quaternions. In order to do so we shall have to fundamentally alter the structure of how these diagrams are constructed: Composition of rotations along different axes is not commutative, something that the spider-mediated monoid of ZX cannot model (§6.1). While there have been graphical calculi with phase groups outside of $([0, 2\pi), +)$ before, such as Backens' calculus representing Spekkens' toy bit theory [BD15], to the author's knowledge the language ZQ is the first with a non-commutative phase group.

To emphasise this idea further: Much study has been done on fragments of ZX [JPV19], but these fragments all have a phase group that sits inside the unit circle of the complex plane. Likewise ZW$_\mathbb{C}$ and ZH$_\mathbb{C}$ have phase rings of the entire complex plane. Indeed we showed in Proposition 5.26 that a phase ring for a qubit graphical calculus must inject into $\mathbb{C}$. ZQ is therefore interesting as a qubit graphical calculus with a phase algebra that does not neatly fit into the complex numbers, quite apart from its practical uses.

The Bloch Sphere, which we cover in more detail in §6.1, is not a perfect analogy. Although it provides us with useful intuition and a way to consider a single qubit in real Euclidean space, its group of rotations, SO3($\mathbb{R}$), is a subgroup of the group of unitary evolutions, SU2($\mathbb{C}$), which the standard model of quantum computing actually uses [NC10]. The group SU2($\mathbb{C}$) itself is isomorphic to the group of unit-length quaternions, and it is this group of quaternions that we shall use as rotations, giving us the 'Q' in ZQ. This use of quaternions to represent rotations is nothing new in the world of engineering and computer graphics [Sho85], but has recently surfaced as a useful component of intermediate representations for quantum circuits. Intermediate representations sit between the user's specification of an algorithm and the actual implementation on a specific piece of hardware. The system TriQ [Mur+19] provides such an intermediate representation, targeting existing quantum computers run by IBM, Rigetti, and the University of Maryland. The authors claim a speed-up in execution of their benchmarks on the seven quantum computers considered, in part because of TriQ's use of quaternions in the optimisation process [Mur+19, §4]: Any sequence of single qubit rotations can be combined into just one quaternion, then decomposed into the most efficient sequence of gates for the target hardware architecture. Our aims in making ZQ are the following:

- Construct a graphical calculus that expresses all qubit rotations

- Construct a qubit graphical calculus with a non-commutative phase group

- Provide a complete ruleset that can condense sequences of single qubit rotations

- Provide a complete ruleset for a qubit phase group calculus that only uses linear side conditions

This last point deserves extra attention: The (EU') rule of ZX, given with the rest of the Universal ZX rules in Figure 6.6, has side conditions that are fiddly to compute, making them unsuitable for researchers to use or remember, and extremely hard to find through conjecture synthesis (we explore this idea further in §7.2). Neither of these objections apply to ZQ, with the Euler Angle composition captured neatly by the multiplication of quaternions. Before we give the definition of ZQ we first give a brief overview of the Bloch Sphere, the groups SU2($\mathbb{C}$) and SO3($\mathbb{R}$), and unit quaternions.

6.1   Quaternions, Rotations and the Bloch Sphere

In this section we will introduce the Bloch Sphere and quaternions, then demonstrate the incompatibility of the group of unit-length quaternions with the spider notation of ZX.

Definition 6.1 (Bloch Sphere [NC10, Figure 1.3]). A qubit state, up to global phase, can be represented as

$$| \psi \rangle = \cos \frac{\theta}{2} | 0 \rangle + e^{i\phi} \sin \frac{\theta}{2} | 1 \rangle \tag{6.1}$$

The parameters θ and ϕ describe a point on the unit sphere in $\mathbb{R}^3$, known as the Bloch Sphere. The group of rotations of this sphere is SO3($\mathbb{R}$).

Definition 6.2 (Quaternions [HGK04, p12]). The quaternions are a non-commutative, four-dimensional, real algebra:

$$\mathbb{R} + i\mathbb{R} + j\mathbb{R} + k\mathbb{R} \tag{6.2}$$
$$i^2 = j^2 = k^2 = ijk = -1 \tag{6.3}$$

For ZQ we are only interested in unit-length quaternions, forming the group $\hat{Q}$ under multiplication.

Remark 6.3. The group $\hat{Q}$ is isomorphic with SU2($\mathbb{C}$), via the isomorphism:

$$\phi : \hat{Q} \to \text{SU2}(\mathbb{C}) \tag{6.4}$$

$$q_w + iq_x + jq_y + kq_z \mapsto \begin{pmatrix} q_w - iq_z & -q_y + iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix} \tag{6.5}$$

The fact that the groups are isomorphic is well known, but we include a proof that this particular map is an isomorphism in Proposition C.1.

There is another way to represent unit-length quaternions, and that is by an angle and a unit vector. It is important to note that this is not the same thing as 'an angle rotation along a unit vector'; the angle-vector pair $(\alpha, \hat{v})$ and the angle-vector pair $(-\alpha, -\hat{v})$ are different as pairs, but would constitute the same rotation. This, in fact, describes the relationship between $\hat{Q}$ and SO3($\mathbb{R}$):

Definition 6.4 (Quaternions as rotations). There is a canonical homomorphism from $\hat{Q}$ to SO3($\mathbb{R}$), given by

$$(\alpha, v) := \cos \frac{\alpha}{2} + \sin \frac{\alpha}{2}(iv_x + jv_y + kv_z) \tag{6.6}$$

$$\psi : \hat{Q} \to \text{SO3}(\mathbb{R}) \tag{6.7}$$
$$(\alpha, v) \mapsto \text{ rotation by angle } \alpha \text{ along vector } v \tag{6.8}$$

$$\ker \psi = \{1, -1\} \tag{6.9}$$

Remark 6.5. This presentation of quaternions as angle-vector pairs is directly linked to the Pauli matrices and the Hadamard map via the homomorphism ϕ of (6.4), up to a scalar factor of $-i$:

$$\phi((\pi, x)) = -i \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{6.10}$$

$$\phi((\pi, y)) = -i \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \tag{6.11}$$

$$\phi((\pi, z)) = -i \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6.12}$$

$$\phi((\pi, \frac{1}{\sqrt{2}}(x + z))) = \frac{-i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{6.13}$$

Generally

$$\phi((\pi, v)) = -i \begin{pmatrix} v_z & v_x + iv_y \\ v_x - iv_y & -v_z \end{pmatrix} \tag{6.14}$$

Definition 6.6 (The H quaternion). The quaternion $(\pi, \frac{1}{\sqrt{2}}(x + z))$ will often just be referred to as H. It corresponds to the rotation that maps the Z axis to the X axis and vice versa.

Example 6.7 (Quaternions in TriQ). The compiler TriQ uses quaternions as rotations as part of its optimisation process[1]:

> Since 1Q operations are rotations, each 1Q gate in the IR can be expressed using a unit rotation quaternion which is a canonical representation using a 4D complex number. TriQ composes rotation operations by multiplying the corresponding quaternions and creates a single arbitrary rotation. This rotation is expressed in terms of the input gate set. Furthermore, on all three vendors, Z-axis rotations are special operations that are implemented in classical hardware and are therefore error-free. TriQ expresses the multiplied quaternion as a series of two Z-axis rotations and one rotation along either X or Y axis, thereby maximizing the number of error-free operations. (Full-Stack, Real-System Quantum Computer Studies: Architectural Comparisons and Design Insights [Mur+19])

---

[1]1Q: One qubit, 2Q: Two qubits, IR: Intermediate representation

We shall explicitly construct this decomposition of a quaternion into a Z-X-Z rotation in Proposition 6.29 when we explore how to translate from ZQ to ZX.

Unit-length quaternions form a very different group to the rotations of just the Z axis. We will now show that the usual presentation of ZX (spiders and states) is fundamentally incompatible with the group $\hat{Q}$. In order to do so we introduce the notion of a generic monoid, similar to how in Definition 5.1 we constructed a generic PROP for rings, RING-PROP$_R$.

Definition 6.8 (Monoid [Mac13, p2]). A monoid is a set M equipped with a binary operation:

$$\mu : M \times M \to M \tag{6.15}$$

referred to as the multiplication, which is required to be associative. There is also a unit, e, which is the left and right unit for μ. We can also view e as the image of a function $\eta : 1 \to M$. Graphically we represent these as:

$$\mu := \qquad\qquad\qquad\qquad \eta := \tag{6.16}$$

such that the following hold:

$$\tag{6.17}$$

Definition 6.9 (A generic monoid PROP). We construct the generic monoid PROP, MONOID-PROP$_M$, for the monoid (M, μ, e) as the PROP generated by

$$\qquad\text{and}\qquad \left\{ \begin{array}{c} \\ a \end{array} \right\}_{a \in M} \tag{6.18}$$

With the following rewrite rules:

$$\underset{unit_l}{=} \qquad \underset{unit_r}{=} \tag{6.19}$$

$$\underset{assoc}{=} \tag{6.20}$$

A model for MONOID-PROP$_M$ in a graphical calculus G is a strict symmetric monoidal functor MONOID-PROP$_M \to G$ (just as in Definition 5.3).

Proposition 6.10. *Every faithful model of* MONOID-PROP$_M$ *into a graphical calculus over* **Qubit** *is commutative.*

Proof. Taking a generic model of MONOID-PROP$_M$ we look at the interpretation of the image of the generators of MONOID-PROP$_M$. For brevity we will just write $[\![\, D \,]\!]$ for D a diagram in MONOID-PROP$_M$ to mean the interpretation of the image of D in the model. We proceed by looking at the span of the interpretations of the elements of M.

$$W := span\left\{ \left[\!\left[\; \underset{e}{\downarrow} \;\right]\!\right], \left[\!\left[\; \underset{a}{\downarrow} \;\right]\!\right], \left[\!\left[\; \underset{a'}{\downarrow} \;\right]\!\right], \dots \right\} \tag{6.21}$$

- If $\dim W = 0$ then the monoid has only one element, e, and so is commutative.

- If $\dim W > 0$ then there either $M = \{e\}$ (and so commutative), or there is some other element $a \in M$. This implies that $\left[\!\left[\; \underset{e}{\downarrow} \;\right]\!\right] \neq 0$:

$$\text{assume} \quad \left[\!\left[\; \underset{e}{\downarrow} \;\right]\!\right] = 0 \tag{6.22}$$

$$\therefore \left[\!\left[\; \underset{\overset{\bigcirc}{e\;\;a}}{\downarrow} \;\right]\!\right] = 0 \quad (\text{any element } a) \tag{6.23}$$

$$\left[\!\left[\; \underset{\overset{\bigcirc}{e\;\;a}}{\downarrow} \;\right]\!\right] = \left[\!\left[\; \underset{a}{\downarrow} \;\right]\!\right] \quad (e \text{ is the unit for } m) \tag{6.24}$$

$$\therefore \left[\!\left[\; \underset{a}{\downarrow} \;\right]\!\right] = \left[\!\left[\; \underset{e}{\downarrow} \;\right]\!\right] \quad \text{⨶} \quad (\text{faithfulness}) \tag{6.25}$$

- If $\dim W = 1$ then without loss of generality:

$$\left[\!\left[\; \underset{e}{\downarrow} \;\right]\!\right] = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{6.26}$$

$$\therefore \left[\!\left[\; \underset{a}{\downarrow} \;\right]\!\right] = \begin{pmatrix} \lambda_a \\ 0 \end{pmatrix} \forall a \quad \text{some } \lambda_a \in \mathbb{C} \tag{6.27}$$

$$\left[\!\left[\; \underset{\overset{\bigcirc}{e\;\;a}}{\downarrow} \;\right]\!\right] = \left[\!\left[\; \underset{a}{\downarrow} \;\right]\!\right] \quad e \text{ is the unit for } m \tag{6.28}$$

$$\therefore \left[\!\left[\; \underset{\wedge}{\overset{\downarrow}{\bigcirc}} \;\right]\!\right] = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot \end{pmatrix} \quad \text{where } \cdot \text{ represents unknowns} \tag{6.29}$$

$$\therefore \left[\!\!\left[\; \vcenter{\hbox{(spider with a, b)}} \; \right]\!\!\right] = \begin{pmatrix} \lambda_a \lambda_b \\ 0 \end{pmatrix} = \left[\!\!\left[\; \vcenter{\hbox{(spider with b, a)}} \; \right]\!\!\right] \tag{6.30}$$

- if $\dim W = 2$ then the states span all of $\mathbb{C}^2$:

$$\left[\!\!\left[\; \vcenter{\hbox{(e state)}} \; \right]\!\!\right] = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{w.l.o.g} \tag{6.31}$$

$$\vcenter{\hbox{(spider e, a)}} \underset{unit_l}{=} \vcenter{\hbox{(a state)}} \quad \forall a \tag{6.32}$$

$$\therefore \left[\!\!\left[\; \vcenter{\hbox{(diagram e left)}} \; \right]\!\!\right] = \left[\!\!\left[\; \vcenter{\hbox{(identity)}} \; \right]\!\!\right] = \left[\!\!\left[\; \vcenter{\hbox{(diagram e right)}} \; \right]\!\!\right] \quad \text{since } span\left\{ \vcenter{\hbox{(a state)}} \right\} = \mathbb{C}^2 \tag{6.33}$$

$$\therefore \left[\!\!\left[\; \vcenter{\hbox{(multiplication)}} \; \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & \cdot \\ 0 & 1 & 1 & \cdot \end{pmatrix} \quad \text{where } \cdot \text{ represents unknowns} \tag{6.34}$$

$$\therefore \left[\!\!\left[\; \vcenter{\hbox{(diagram)}} \; \right]\!\!\right] = \left[\!\!\left[\; \vcenter{\hbox{(diagram)}} \; \right]\!\!\right] \tag{6.35}$$

$$\therefore \left[\!\!\left[\; \vcenter{\hbox{(spider a, b)}} \; \right]\!\!\right] = \left[\!\!\left[\; \vcenter{\hbox{(spider b, a)}} \; \right]\!\!\right] \tag{6.36}$$

$\square$

Corollary 6.11. *There is no qubit graphical calculus that faithfully models* MONOID-PROP$_M$ *(for any M) such that this monoid action is isomorphic to the group multiplication of $\hat{Q}$, because $\hat{Q}$ is not commutative.*

Remark 6.12. Since $\hat{Q}$ cannot be modelled as a monoid over states, it cannot obey the fundamental spider rules of [CD08]. We therefore note that the presentation of ZQ (when we come to it) will be different by necessity rather than whim.

We will deviate from the structure of ZX by having *labelled, directed edges* carry the phase group of ZQ, but keep phase-free Z spiders to mediate the graph structure. Although ZQ is presented here in the form of a PROP we encourage the reader to think of ZQ diagrams as directed, multi, open graphs, where each directed edge is labelled by a unit-length quaternion. A different approach to this monoid-based presentation of groups

(as suggested by our general notion of graphical calculi based on Σ-algebras in §5.4) is to instead consider a group as a category with one object and where all morphisms are invertible. This point of view lends us to the following definition:

Definition 6.13 (Group PROP with vertical composition)**.** We construct the (vertically composed) group PROP, V-GROUP-PROP$_G$, for a group $(G, \cdot)$ as the PROP with generators

$$
\begin{array}{c}
\text{\large g} \quad \forall g \in G
\end{array}
\tag{6.37}
$$

It additionally satisfies the rewrite rules

$$
\begin{array}{c}
\text{\large f} \\
\text{\large g}
\end{array}
\ \ _{\bar{\bar{M}}} \ \
\boxed{f \cdot g}
\tag{6.38}
$$

Both ZQ and ZX will be models of this PROP; in each case the image of the morphisms in V-GROUP-PROP$_G$ are the single qubit rotations of the phase group. For ZQ this group is SU2($\mathbb{C}$), for ZX this group is $[0, 2\pi)$ under addition.

6.2   The definition of the ZQ-calculus

We present the graphical calculus ZQ as a compact closed PROP generated by the morphisms in Figure 6.1 and then present the interpretation of these generators in Figure 6.3. We build the transpose of the Q_q node in the usual way, as shown in Figure 6.2. A unit-length quaternion q can be expressed as an element of $\mathbb{R}^4$, i.e. $q_w + iq_x + jq_y + kq_z$, or as an angle-vector pair, e.g. (α, v) (see §6.1). Finally we give the rules of ZQ in Figure 6.5.

Definition 6.14 (ZQ)**.** The graphical calculus ZQ is formed by:

- The generators of Figure 6.1

- The interpretation of Figure 6.3

- The rules of Figure 6.5

Theorem 6.15 (ZQ is sound)**.** *The rules of ZQ (Figure 6.5) are sound with respect to the standard interpretation (Figure 6.3)*

Proof. This proof is covered in §6.3 $\square$

Theorem 6.16 (ZQ is complete)**.** *ZQ with the rules of Figure 6.5 is complete with respect to the standard interpretation (Figure 6.3)*

$$Z : m \to n \qquad \text{Z spider node}$$

$$Q_q : 1 \to 1 \qquad q \text{ a unit-length quaternion}$$

$$\lambda_c : 0 \to 0 \qquad \lambda_c \qquad c \text{ a complex scalar}$$
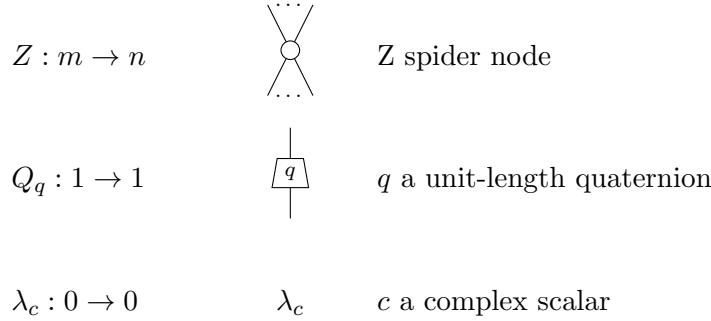
Figure 6.1: The generators of ZQ as a PROP
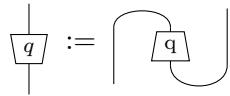
Figure 6.2: The transpose of q in ZQ

Proof. This proof is covered in §6.4, and is performed by an equivalence with the ZX calculus. □

Diagrams of ZQ are elements of the compact closed PROP generated by the morphisms in Figure 6.1. We add to these generators a shorthand for the transpose of Q_q, written as a trapezium pointing downwards (as in Figure 6.2), and defined using the cup and cap. The interpretations of these generators are given both in Figure 6.3, or alternatively using angle-vector pair notation (Definition 6.4) we have the interpretation given in Figure 6.4.

$$\left[\!\!\left[\ \rangle\!\!\langle\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & \dots & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & \dots & 0 & 0 \\ 0 & & \dots & 0 & 1 \end{pmatrix}$$

$$\left[\!\!\left[\ \boxed{q}\ \right]\!\!\right] = \begin{pmatrix} q_w - iq_z & -q_y + iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix}$$

$$\left[\!\!\left[\ \lambda_c\ \right]\!\!\right] = c$$

$$\left[\!\!\left[\ \cup\ \right]\!\!\right] = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\left[\!\!\left[\ \cap\ \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Figure 6.3: Interpretations of the generators of ZQ

$$\left[\!\!\left[\ \boxed{(\alpha, v)}\ \right]\!\!\right] = \begin{pmatrix} \cos\frac{\alpha}{2} - i\sin\frac{\alpha}{2}v_z & -i\sin\frac{\alpha}{2}(v_x + iv_y) \\ -i\sin\frac{\alpha}{2}(v_x - iv_y) & \cos\frac{\alpha}{2} + i\sin\frac{\alpha}{2}v_z \end{pmatrix}$$

Figure 6.4: Interpretation of the Q generator using angle-vector pair notation

In order to decrease diagrammatic clutter we shall use the following notation:

$$\vdots \ := \ \boxed{(\pi, \tfrac{1}{\sqrt{2}}(x+z))} \ = \ \boxed{H} \tag{6.39}$$

This is the familiar 'Hadamard edge' from e.g. [Dun+20]. Note that the Hadamard edge is symmetrical, but the H quaternion edge decoration is not. As soon as we have the tools to show that this is well defined we shall do so (Lemma 6.17).

6.2.1   The rules of ZQ

We present in Figure 6.5 a ruleset of ZQ. The soundness of this ruleset is proved in subsection 6.3 and completeness is proved in section 6.4.

Lemma 6.17. *The Hadamard edge is well defined in ZQ, in that:*

$$ZQ \vDash \boxed{H} \ = \ \boxed{H} \ = \ \vdots \qquad\qquad ZQ \vdash \boxed{H} \ \overset{=}{Y} \ \boxed{H} \ = \ \vdots \tag{6.40}$$

Proof. For the semantics:

$$\left[\!\!\left[\ \boxed{H}\ \right]\!\!\right] = \frac{-i}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \left[\!\!\left[\ \boxed{H}\ \right]\!\!\right]^{T} = \left[\!\!\left[\ \left(\boxed{H}\right)^{T}\ \right]\!\!\right] = \left[\!\!\left[\ \boxed{H}\ \right]\!\!\right] \tag{6.41}$$

Syntactically

$$\boxed{\cos\tfrac{\pi}{2} + \sin\tfrac{\pi}{2}(i+k)} \ \overset{=}{Y} \ \boxed{\cos\tfrac{\pi}{2} + \sin\tfrac{\pi}{2}(i+k)} \tag{6.42}$$

$\square$

The author feels that the rules of ZQ are easier to grasp than the rules of Universal ZX given in Figure 6.6. (Other rulesets for Universal ZX exist but all of them require trigonometric side conditions for their variation on the EU rule.) By contrast the rules of ZQ involve a single, linear side condition, it abstracts scalar diagrams into simply scalar numbers λ_c, and it neatly divides graphically into entanglement (vertices) and rotations (edges). The 'complexity' of the (EU') rule in ZX is simply a shadow of the group multiplication of rotations shown in the (Q) rule of ZQ. This is best illustrated by noting that to simplify a chain of single qubit unitaries in ZX one must successively apply the (EU') rule, at each point calculating x^{+}, x^{-}, z, and z', as well as various arguments, moduli, sines, and cosines, eventually reaching three rotations (and a choice of ZXZ or XZX); in ZQ one simply applies the group multiplication.
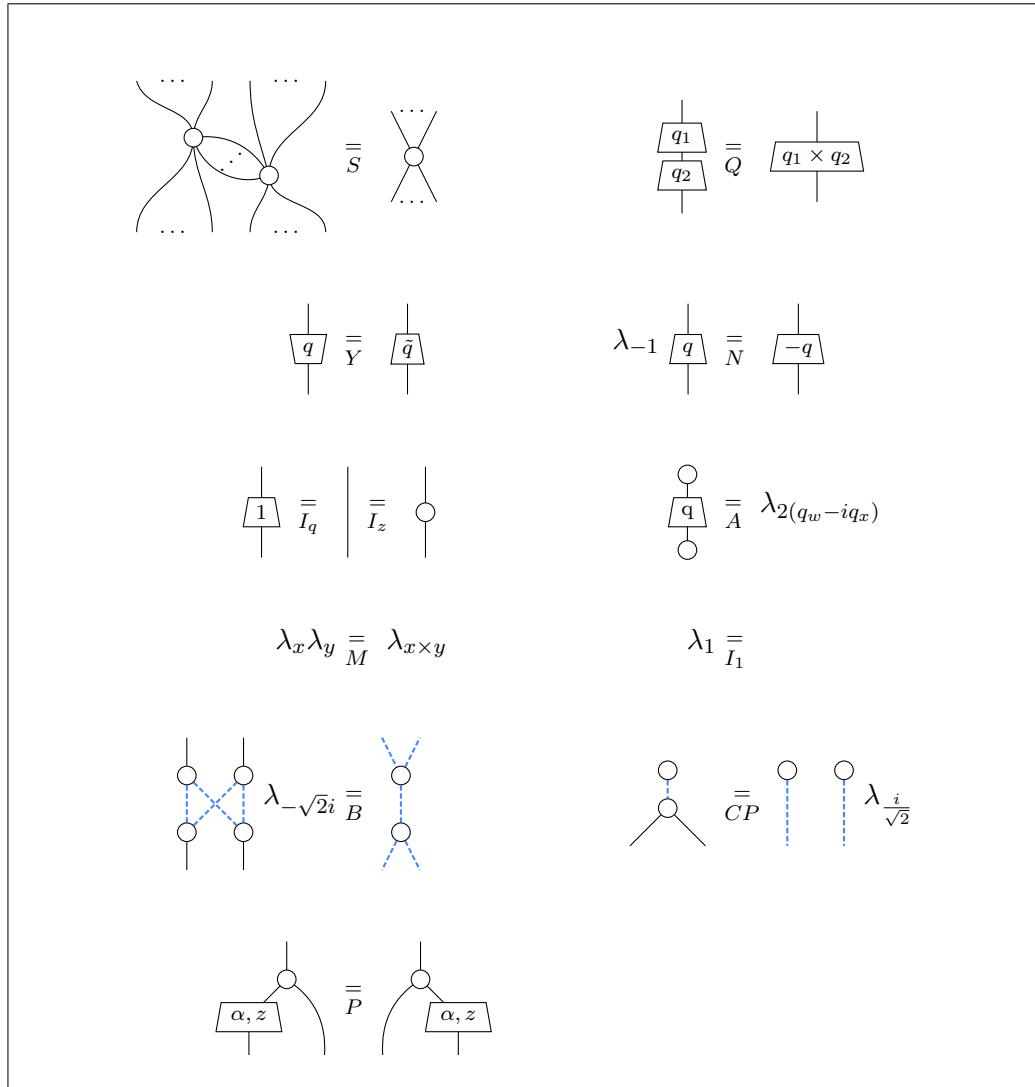
Figure 6.5: The rules of ZQ. In rule S the diagonal dots indicate one or more wires, horizontal dots indicate zero or more wires. The right hand side of rule I_1 is the empty diagram, and $\tilde{q}$ is the quaternion q reflected in the map $j \mapsto -j$.
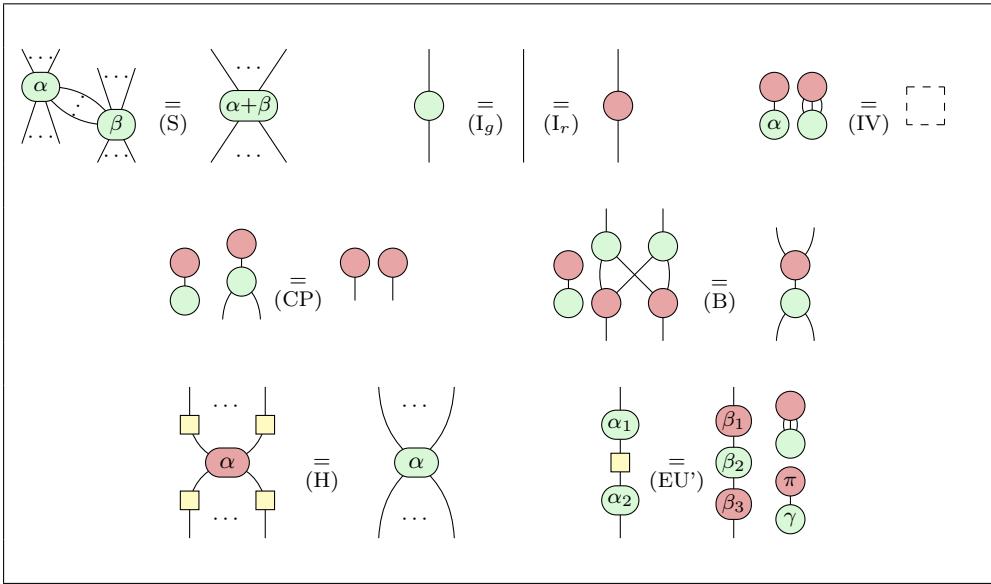
Figure 6.6: Set of rules ZX for the ZX-Calculus with scalars from Ref. [Vil19]. The right-hand side of (IV) is an empty diagram. (...) denote zero or more wires, while ($\cdot\cdot$) denote one or more wires. In rule (EU'), $\beta_1, \beta_2, \beta_3$ and γ can be determined as follows: $x^+ := \frac{\alpha_1 + \alpha_2}{2}$, $x^- := x^+ - \alpha_2$, $z := -\sin x^+ + i\cos x^-$ and $z' := \cos x^+ - i\sin x^-$, then $\beta_1 = \arg z + \arg z', \beta_2 = 2\arg\left(i + \left|\frac{z}{z'}\right|\right), \beta_3 = \arg z - \arg z', \gamma = x^+ - \arg(z) + \frac{\pi - \beta_2}{2}$ where by convention $\arg(0) := 0$ and $z' = 0 \implies \beta_2 = 0$.

6.3   Soundness of ZQ

In this section we go through each of the rules given in Section 6.2.1, showing that the interpretations of the left and right hand sides of the rules are equal.

Proposition 6.18. *The rule S is sound:*

$$\left[\!\!\left[\vcenter{\hbox{\includegraphics{lhs}}}\right]\!\!\right] = \left[\!\!\left[\vcenter{\hbox{\includegraphics{rhs}}}\right]\!\!\right] \tag{6.43}$$

Where there are $k \geq 1$ wires represented by $\cdot\cdot$ in the middle of the left hand side.

Proof. This is simply a restating of the original Z spider law from [CD08, Theorem 6.12]. □

Proposition 6.19. *The rule Q is sound:*

$$\left[\!\!\left[\begin{array}{c}\boxed{q_1}\\\boxed{q_2}\end{array}\right]\!\!\right] = \left[\!\!\left[\;\boxed{q_1 \times q_2}\;\right]\!\!\right] \tag{6.44}$$

Proof. Follows from ϕ (see Remark 6.3) being a group isomorphism. The left hand side is multiplication in $SU2(\mathbb{C})$, the right hand side is multiplication in $\hat{Q}$. $\qquad\square$

Proposition 6.20. *The rule Y is sound:*

$$\left[\!\!\left[\;\boxed{q_w + iq_x - jq_y + kq_z}\;\right]\!\!\right] = \left[\!\!\left[\;\boxed{q_w + iq_x + jq_y + kq_z}\;\right]\!\!\right] \tag{6.45}$$

Proof. The action of the cups and caps in Figure 6.2 (where we defined the diagrammatic transpose), is to enact the transpose in the interpretation:

$$\left[\!\!\left[\;\boxed{q}\;\right]\!\!\right] = \begin{pmatrix} q_w - iq_z & -q_y - iq_x \\ q_y - iq_x & q_w + iq_z \end{pmatrix} = \left[\!\!\left[\;\boxed{q_w + iq_x - jq_y + kq_z}\;\right]\!\!\right] \tag{6.46}$$

$$\square$$

Proposition 6.21. *The rule N is sound:*

$$\left[\!\!\left[\;\lambda_{-1}\;\boxed{q}\;\right]\!\!\right] = \left[\!\!\left[\;\boxed{-q}\;\right]\!\!\right] \tag{6.47}$$

Proof.

$$LHS = -1 \begin{pmatrix} q_w - iq_z & q_y - iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix} = \begin{pmatrix} -q_w + iq_z & -q_y + iq_x \\ q_y + iq_x & -q_w - iq_z \end{pmatrix} = RHS \tag{6.48}$$

$$\square$$

Proposition 6.22. *The rules I_q and I_z are sound:*

$$\left[\!\!\left[\;\boxed{1}\;\right]\!\!\right] = \left[\!\!\left[\;\Big|\;\Big|\;\right]\!\!\right] = \left[\!\!\left[\;\bigcirc\;\right]\!\!\right] \tag{6.49}$$

Proof. They all have the interpretation $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. $\qquad\square$

Proposition 6.23. *The rule A is sound:*

$$\left[\!\!\left[\begin{array}{c} \bigcirc \\ \boxed{q} \\ \bigcirc \end{array} \right]\!\!\right] = \left[\!\!\left[\; \lambda_{2(q_w - iq_x)} \; \right]\!\!\right] \tag{6.50}$$

Proof.

$$\left[\!\!\left[\begin{array}{c} \bigcirc \\ \boxed{q} \\ \bigcirc \end{array} \right]\!\!\right] = \begin{pmatrix} 1 & 1 \end{pmatrix} \circ \begin{pmatrix} q_w - iq_z & q_y - iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{6.51}$$

$$= q_w - iq_z + q_y - iq_x - q_y - iq_x + q_w + iq_z \tag{6.52}$$

$$= 2(q_w - iq_x) \tag{6.53}$$

$$= \left[\!\!\left[\; \lambda_{2(q_w - iq_x)} \; \right]\!\!\right] \tag{6.54}$$

$\square$

Proposition 6.24. *The rule M is sound:*

$$\left[\!\!\left[\; \lambda_x \lambda_y \; \right]\!\!\right] = \left[\!\!\left[\; \lambda_{x \times y} \; \right]\!\!\right] \tag{6.55}$$

Proof. Both sides have interpretation $x \times y$. $\square$

Proposition 6.25. *The rule I_λ is sound:*

$$\left[\!\!\left[\; \lambda_1 \; \right]\!\!\right] = \left[\!\!\left[\; \epsilon \; \right]\!\!\right] \tag{6.56}$$

Where ϵ is the empty diagram.

Proof. Both sides have interpretation 1. $\square$

Proposition 6.26. *The rule B is sound:*

$$\left[\!\!\left[\begin{array}{c} \lambda_{-\sqrt{2}i} \end{array} \right]\!\!\right] = \left[\!\!\left[\right]\!\!\right] \tag{6.57}$$

Proof.

$$LHS = -\sqrt{2}i \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{\otimes 2} \circ \left(\frac{-i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right)^{\otimes 4} \circ \tag{6.58}$$

$$\left(id_2 \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes \mathrm{id}_2 \right) \circ \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes 2} \tag{6.59}$$

$$= \frac{-i}{\sqrt{2}^3} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{6.60}$$

$$RHS = \left(\frac{-i}{\sqrt{2}} \right)^5 \times \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes 2} \circ \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \circ \tag{6.61}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes 2} \tag{6.62}$$

$$= \left(\frac{-i}{\sqrt{2}} \right)^5 \times \begin{pmatrix} 2 & 2 & 2 & 2 \\ 2 & -2 & -2 & 2 \\ 2 & -2 & -2 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix} = \frac{-i}{\sqrt{2}^3} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{6.63}$$

$\square$

Proposition 6.27. *The rule CP is sound:*

$$\left[\!\!\left[\begin{array}{c} \ \end{array} \right]\!\!\right] = \left[\!\!\left[\begin{array}{cc} \ & \ \end{array} \lambda_{\frac{i}{\sqrt{2}}} \right]\!\!\right] \tag{6.64}$$

Proof.

$$LHS = \begin{pmatrix} 1 & 1 \end{pmatrix} \circ \frac{-i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \frac{-i}{\sqrt{2}} \begin{pmatrix} 2 & 0 & 0 & 0 \end{pmatrix} \tag{6.65}$$

$$RHS = \frac{i}{\sqrt{2}} \left(\begin{pmatrix} 1 & 1 \end{pmatrix} \circ \frac{-i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right)^{\otimes 2} = \frac{-i}{\sqrt{2}} \begin{pmatrix} 2 & 0 & 0 & 0 \end{pmatrix} \tag{6.66}$$

$\square$

Proposition 6.28. *The rule P is sound:*

$$\left[\!\!\left[\;\raisebox{-0.5em}{}\; \right]\!\!\right] = \left[\!\!\left[\;\raisebox{-0.5em}{}\; \right]\!\!\right] \tag{6.67}$$

Proof.

$$LHS = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \circ \left(\begin{pmatrix} q_w - iq_z & 0 \\ 0 & q_w + iq_z \end{pmatrix} \otimes id_2 \right) \tag{6.68}$$

$$= \begin{pmatrix} q_w - iq_z & 0 & 0 & 0 \\ 0 & 0 & 0 & q_w + iq_z \end{pmatrix} \tag{6.69}$$

$$RHS = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \circ \left(id_2 \otimes \begin{pmatrix} q_w - iq_z & 0 \\ 0 & q_w + iq_z \end{pmatrix} \right) \tag{6.70}$$

$$= \begin{pmatrix} q_w - iq_z & 0 & 0 & 0 \\ 0 & 0 & 0 & q_w + iq_z \end{pmatrix} \tag{6.71}$$

□

6.4   Completeness of ZQ

The completion of ZQ is achieved by finding an equivalence between ZQ and ZX as PROPs. We already know that ZX is complete [NW17] and this proof was by a similar equivalence with ZW, which was shown to be complete in Ref. [Had15]. Equivalence is shown by finding a translation of the generators from ZX to ZQ and vice versa (§6.4.1), before then translating all of the rules from ZX into ZQ (§6.4.2), and keeping these as rules in ZQ. Finally one has to ensure that any diagram translated from ZQ to ZX and back again can be proven to be equivalent to the original ZQ diagram (§6.4.3). In symbols this is:

$$[\![D_1]\!] = [\![D_2]\!] \quad \text{Two diagrams in ZQ} \tag{6.72}$$

$$ZX \vdash F_X D_1 = F_X D_2 \tag{6.73}$$

$$\S 6.4.2: ZQ \vdash F_Q F_X D_1 = F_Q F_X D_2 \tag{6.74}$$

$$\S 6.4.3: ZQ \vdash D_1 = F_Q F_X D_1 \quad \text{and} \quad D_2 = F_Q F_X D_2 \tag{6.75}$$

$$\therefore ZQ \vdash D_1 = F_Q F_X D_1 = F_Q F_X D_2 = D_2 \tag{6.76}$$

6.4.1   Translation to and from ZX

We define the strict monoidal functors F_X and F_Q on generators in Figure 6.7. In defining this translation we make use of two facts; firstly that we can decompose any unit quaternion into Z then X then Z rotations. This is tantamount to Euler Angle Decomposition and is performed explicitly in Proposition 6.29. Secondly we need to be able to express any complex number in a rather particular form, which is shown in Lemma 6.30.

Proposition 6.29. *There exist α and $\gamma \in [0, 2\pi)$, and $\beta \in [0, \pi]$ such that:*

$$q_w + iq_x + jq_y + kq_z = \left(\cos\frac{\alpha}{2} + k\sin\frac{\alpha}{2}\right)\left(\cos\frac{\beta}{2} + i\sin\frac{\beta}{2}\right)\left(\cos\frac{\gamma}{2} + k\sin\frac{\gamma}{2}\right) \quad (6.77)$$

Proof.

$$RHS = \left(\cos\frac{\alpha}{2} + k\sin\frac{\alpha}{2}\right)\left(\cos\frac{\beta}{2} + i\sin\frac{\beta}{2}\right)\left(\cos\frac{\gamma}{2} + k\sin\frac{\gamma}{2}\right) \quad (6.78)$$

$$= \left(\cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} - \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\sin\frac{\gamma}{2}\right) + \quad (6.79)$$

$$i\left(\cos\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2}\right)$$

$$j\left(\cos\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} - \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2}\right) +$$

$$k\left(\cos\frac{\alpha}{2}\cos\frac{\beta}{2}\sin\frac{\gamma}{2} + \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}\right) +$$

$$= \cos\frac{\beta}{2}\left(\cos\frac{\alpha+\gamma}{2} + i\sin\frac{\alpha+\gamma}{2}\right) + j\sin\frac{\beta}{2}\left(\sin\frac{\gamma-\alpha}{2} - i\cos\frac{\gamma-\alpha}{2}\right)$$

From this we gather:

$$q_w = \cos\frac{\beta}{2}\cos\frac{\alpha+\gamma}{2} \qquad\qquad q_x = \cos\frac{\beta}{2}\sin\frac{\alpha+\gamma}{2} \qquad (6.80)$$

$$q_y = \sin\frac{\beta}{2}\sin\frac{\gamma-\alpha}{2} \qquad\qquad q_z = \sin\frac{\beta}{2}\cos\frac{\gamma-\alpha}{2} \qquad (6.81)$$

And finally use these to determine values of α, β and γ :

- $q_w^2 + q_x^2 = \cos^2\frac{\beta}{2}$ determines up to two different possibilities of $\beta \in [0, 2\pi)$. We will enforce $\beta \in [0, \pi]$ to make this unique and $\cos\frac{\beta}{2}$ non-negative.

- If $\beta = 0$ then set $\gamma = 0$, use q_w and q_x to determine α

- Likewise if $\beta = \pi$ set $\gamma = 0$, use q_y and q_z to determine α

- Otherwise determine $\alpha + \gamma/2$ from q_w and q_x, and $\alpha - \gamma/2$ from q_y and q_z; their sum and difference give γ and α respectively.
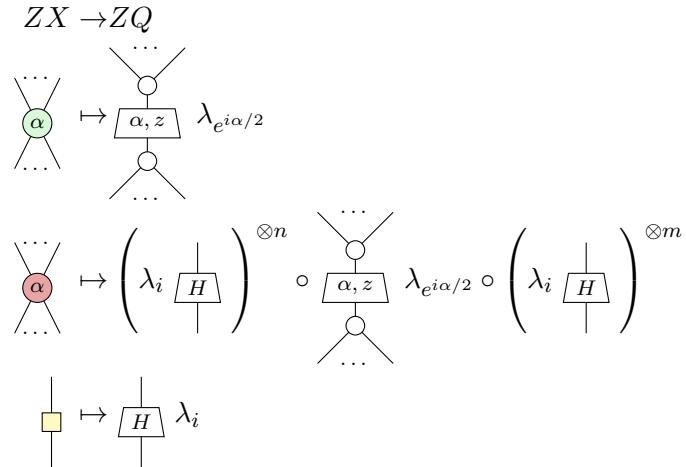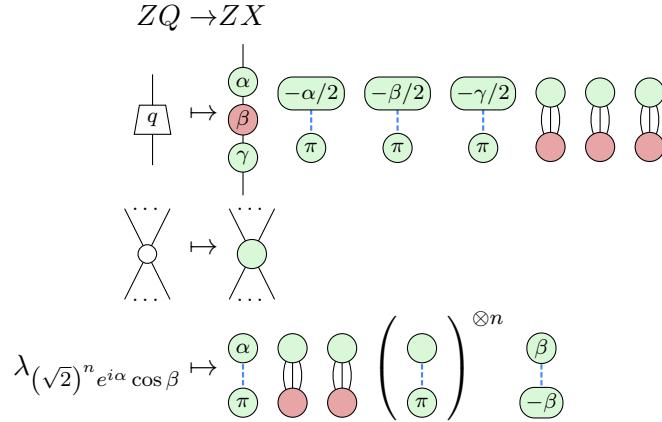
Figure 6.7: Translation from ZX to ZQ and back again. The existence of α, β and γ when translating the Q node is shown in Proposition 6.29, likewise the decomposition of any complex number as $\left(\sqrt{2}\right)^n e^{i\alpha}\cos\beta$ is shown in Lemma 6.30.

The choices we made in this proof we justify by noting that we can represent these choices by certain applications of the spider rule (in the case $\beta = 0$) and π-commutativity rules (relating $(\alpha, \beta, \gamma) \sim (\alpha + \pi, -\beta, \gamma + \pi)$) in ZX. $\qquad\qquad\square$

Lemma 6.30. *Any non-zero complex number c can be expressed uniquely as $\left(\sqrt{2}\right)^n e^{i\alpha} \cos\beta$ where $n \in \mathbb{N}$, $\alpha \in [0, 2\pi)$, $\beta \in [0, \pi)$ and where n is chosen to be the least n such that $\sqrt{2}^n \geq |c|$.*

Proof. Express the complex number c as $re^{i\alpha}$, where $r \in \mathbb{R}_{\geq 0}$. This matches our choice of $\alpha \in [0, 2\pi)$. For all r there is at least one n where $\sqrt{2}^n \geq r$ and so we can find a least such n. Once we know n there is a unique $\beta \in [0, \pi)$ such that $\cos\beta\sqrt{2}^n = r$. $\qquad\square$

If c is zero then we set α and n to 0 and β to $\pi/2$.

6.4.2   Proving the translated ZX rules

We aim to show that the rules translated from ZX are all derivable by the rules in §6.2.1, which we will refer to as ZQ. We will use the ZX ruleset from [Vil19, Figure 2], and refer to individual ZX rules as $\text{ZX}_{\text{rule name}}$. To save space, we will assume applications of the M rule (scalar multiplication) in the statements of the propositions.

Lemma 6.31. *Translation of the Z spider*

$$ZQ \vdash F_Q \left(\vcenter{\hbox{}} \right) = \vcenter{\hbox{}} \tag{6.82}$$

Proof.

$$LHS = \vcenter{\hbox{}} \;\overset{=}{\underset{I_q}{}}\; \vcenter{\hbox{}} \;\overset{=}{\underset{S}{}}\; \vcenter{\hbox{}} \tag{6.83}$$

$\qquad\qquad\square$

Proposition 6.32. *Translation of the Z spider rule*

$$ZQ \vdash F_Q\left(ZX_S\right) \tag{6.84}$$

$$i.e. \quad ZQ \vdash \qquad = \qquad \qquad (6.85)$$

(The diagonal dots represent at least one wire between the Z spiders)

Proof.

$$\overset{=}{S} \qquad (6.86)$$

$$\overset{=}{P,Y} \qquad \overset{=}{Q,S} \qquad (6.87)$$

$$\overset{=}{P,\overline{Q},S} \qquad (6.88)$$

□

Proposition 6.33. *Translation of the Z spider identity*

$$ZQ \vdash F_Q\left(ZX_{I_g}\right) \tag{6.89}$$

$$i.e. \quad ZQ \vdash \quad \phi \circ \boxed{1} \circ \phi \;=\; \big| \tag{6.90}$$

Proof.

$$\phi \circ \boxed{1} \circ \phi \;\overset{=}{I_z}\; \boxed{1} \;\overset{=}{I_q}\; \big| \tag{6.91}$$

$\square$

Proposition 6.34. *Translation of the X spider identity*

$$ZQ \vdash F_Q\left(ZX_{I_r}\right) \tag{6.92}$$

$$i.e. \quad ZQ \vdash \quad \boxed{H} \circ \phi \circ \boxed{1} \circ \phi \circ \boxed{H}\, \lambda_i\lambda_i \;=\; \big| \tag{6.93}$$

Proof.

$$LHS = \boxed{H} \circ \phi \circ \boxed{1} \circ \phi \circ \boxed{H}\, \lambda_i\lambda_i \tag{6.94}$$

$$\overset{=}{I_z}\; \boxed{H} \circ \boxed{1} \circ \boxed{H}\, \lambda_i\lambda_i \tag{6.95}$$

$$\overset{=}{I_q}\; \boxed{H} \circ \boxed{H}\, \lambda_i\lambda_i \;\overset{=}{Q}\; \boxed{-1}\, \lambda_i\lambda_i \tag{6.96}$$

$$\overset{=}{N}\; \boxed{1}\, \lambda_{-1}\lambda_i\lambda_i \;\overset{=}{I_q}\; \big|\, \lambda_{-1}\lambda_i\lambda_i \;\overset{=}{M}\; \big|\, \lambda_1 \;\overset{=}{I_1}\; \big| \tag{6.97}$$

$\square$

We introduce our first three intermediate lemmas, corresponding to properties of the following three ZX diagrams:

$$\text{(diagram)} \;,\; \text{(diagram)} \;,\; \text{(diagram)} \tag{6.98}$$

Lemma 6.35. *Interaction of a Z state and Z effect joined by a Hadamard*

$$ZQ \vdash \quad \begin{array}{c} \boxed{\alpha, z} \\ \vdots \\ \boxed{\beta, z} \end{array} \quad = \lambda_{-\sqrt{2}\left(\left(\sin\frac{\alpha+\beta}{2}\right)+i\cos\frac{\alpha-\beta}{2}\right)} \tag{6.99}$$

Proof.

$$\begin{array}{c} \boxed{\alpha, z} \\ \vdots \\ \boxed{\beta, z} \end{array} \quad \overset{=}{\underset{I_z}{=}} \quad \begin{array}{c} \boxed{\alpha, z} \\ \boxed{H} \\ \boxed{\beta, z} \end{array} \quad \overset{=}{\underset{Q}{=}} \quad \boxed{(\alpha, z) \times H \times (\beta, z)} \quad \overset{=}{\underset{A}{=}} \quad \lambda_{-\sqrt{2}\left(\left(\sin\frac{\alpha+\beta}{2}\right)+i\cos\frac{\alpha-\beta}{2}\right)} \tag{6.100}$$

Since $(\cos\frac{\alpha}{2} + k\sin\frac{\alpha}{2}) \times H \times (\cos\frac{\beta}{2} + k\sin\frac{\beta}{2}) =$

$$\frac{1}{\sqrt{2}}\left(-\sin\frac{\alpha+\beta}{2} + i\left(\cos\frac{\alpha-\beta}{2}\right) + j\left(\sin\frac{\alpha-\beta}{2}\right) + k\left(\cos\frac{\alpha+\beta}{2}\right)\right) \tag{6.101}$$

$\square$

Lemma 6.36. *Interaction of two Hadamard rotations*

$$ZQ \vdash \boxed{H} \circ \boxed{H} = \lambda_{-1} \Big| \tag{6.102}$$

Proof.

$$\boxed{H} \circ \boxed{H} \quad \overset{=}{\underset{Q}{=}} \quad \boxed{H \times H} \quad = \quad \boxed{-1} \quad \overset{=}{\underset{N}{=}} \quad \boxed{1} \lambda_{-1} \overset{=}{\underset{I_q}{=}} \Big| \lambda{-1} \tag{6.103}$$

$\square$

Lemma 6.37. *The value of the scalar describing three Hadamard rotations in parallel*

$$ZQ \vdash \quad \begin{array}{c} \circ \\ \vdots \\ \circ \end{array} \quad = \lambda_{\frac{i}{\sqrt{2}}} \tag{6.104}$$

Proof.

$$\underset{S,\ \overline{6.36}}{=}\ \lambda_{-1} \qquad\qquad \underset{Y,\overline{S}}{=}\ \lambda_{-1} \qquad\qquad (6.105)$$

$$\underset{\overline{M},B}{=}\ \lambda_{-i/\sqrt{2}} \qquad\qquad \underset{Y,\overline{S},I_z}{=}\ \lambda_{-i/\sqrt{2}} \qquad\qquad (6.106)$$

$$\underset{C\overline{P},Y}{=}\ \lambda_{i/2\sqrt{2}} \qquad\qquad \underset{I_z}{=}\ \lambda_{i/2\sqrt{2}} \qquad\qquad (6.107)$$

$$\underset{M,\overline{6.36}}{=}\ \lambda_{-i/2\sqrt{2}} \qquad\qquad \underset{A}{=}\ \lambda_{-i/2\sqrt{2}}\ \lambda_{-i\sqrt{2}}\lambda_{-i\sqrt{2}} \qquad\qquad (6.108)$$

$$\underset{M}{=}\ \lambda_{\frac{i}{\sqrt{2}}} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (6.109)$$

$$\square$$

Proposition 6.38. *Translation of the IV rule*

$$ZQ \vdash F_Q(ZX_{IV}) \qquad\qquad (6.110)$$

$$i.e. \quad ZQ \vdash \lambda_{e^{i\frac{\alpha}{2}}} \quad \boxed{\text{diagram}} = \epsilon \qquad (6.111)$$

Proof.

$$\underset{I_z, I_q, S, M}{=} \qquad (6.112)$$

$$\underset{6.35,\,6.37}{=} \quad \lambda_{e^{i\frac{\alpha}{2}}} \lambda_{-\sqrt{2}ie^{-i\frac{\alpha}{2}}} \lambda_{i/\sqrt{2}} \qquad (6.113)$$

$$\underset{M}{=} \quad \lambda_1 \underset{I_1}{=} \epsilon \qquad (6.114)$$

$\square$

Proposition 6.39. *Translation of the CP rule*

$$ZQ \vdash F_Q (ZX_{CP}) \qquad (6.115)$$

$$i.e. \quad ZQ \vdash \lambda_{-1} \boxed{\text{diagram}} = \boxed{\text{diagram}} \lambda_{-1} \qquad (6.116)$$

Proof.

$$LHS \underset{1_z,\,1_q}{=} \quad \lambda_{-1} \boxed{\text{diagram}} \quad \underset{A}{=} \quad \lambda_{-1}\lambda_{-i\sqrt{2}} \boxed{\text{diagram}} \qquad (6.117)$$

$$\underset{CP}{=} \quad \lambda_{-1}\lambda_{-i\sqrt{2}}\lambda_{i/\sqrt{2}} \boxed{\text{diagram}} \quad \underset{M}{=} \quad \lambda_{-1} \boxed{\text{diagram}} = RHS \qquad (6.118)$$

$\square$

Proposition 6.40. *Translation of the B rule*

$$ZQ \vdash F_Q(ZX_B) \tag{6.119}$$

i.e. $ZQ \vdash \lambda_{-i}$ λ_{-i} (6.120)

Proof.

$$LHS \underset{I_z, I_q}{=} \lambda_{-i} \quad \underset{B}{=} \lambda_{-i}\lambda_{i/\sqrt{2}} \tag{6.121}$$

$$\underset{A, \overline{M}, 6.36}{=} \lambda_{-i} \tag{6.122}$$

□

Proposition 6.41. *Translation of the H rule*

$$ZQ \vdash F_Q(H) \tag{6.123}$$

i.e. $ZQ \vdash \left(\lambda_i \boxed{H}\right)^{\otimes n} \circ \left(\lambda_i \boxed{H}\right)^{\otimes n} \circ \boxed{\alpha, z}\ \lambda_{e^{i\alpha/2}} \circ$ (6.124)

$$\left(\lambda_i \boxed{H}\right)^{\otimes m} \circ \left(\lambda_i \boxed{H}\right)^{\otimes m}$$

$$= \boxed{\alpha, z} \; \lambda_{e^{i\alpha/2}} \tag{6.125}$$

Proof.

$$LHS \underset{6.36}{=} (\lambda_i \lambda_i \lambda_{-1})^{\otimes n} (\lambda_i \lambda_i \lambda_{-1})^{\otimes m} \boxed{\alpha, z} \; \lambda_{e^{i\alpha/2}} \underset{M}{=} \boxed{\alpha, z} \; \lambda_{e^{i\alpha/2}} \tag{6.126}$$

$\square$

Before proving the translation of the (EU') rule (Proposition 6.46) we introduce some helpful lemmas. For the more trigonometric lemmas we delay their proof until Appendix C, because the details of those proofs are not very informative with respect to proving $ZQ \vdash F_Q(EU')$. We reproduce the side conditions for the rule $ZX_{EU'}$ for reference here:

In rule (EU'), β_1, β_2, β_3 and γ can be determined as follows: $x^+ := \frac{\alpha_1 + \alpha_2}{2}$, $x^- := x^- \alpha_2$, $z := -\sin(x^+) + i\cos(x^-)$ and $z' := \cos(x^+) - i\sin(x^-)$, then $\beta_1 = \arg z + \arg z'$, $\beta_2 = 2\arg(i + \frac{|z|}{|z'|})$, $\beta_3 = \arg z - \arg z'$, $\gamma = x^+ - \arg(z) + \frac{\pi - \beta_2}{2}$ where by convention $\arg(0) := 0$ and $z' = 0 \implies \beta_2 = 0$. (Figure 2, A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics [Vil19])

Lemma 6.42. *With the conditions of $ZX_{EU'}$*

$$\left(e^{i(\beta_1 + \beta_2 + \beta_3 + \gamma + 9\pi)/2}\right) \times \left(\frac{i}{\sqrt{2}}\right) \times \left(-\sqrt{2}\left(e^{i\gamma/2}\right)\right) = e^{i(\alpha_1 + \alpha_2 + \pi)/2} \tag{6.127}$$

Proof.

$$LHS = \left(e^{i(\beta_1 + \beta_2 + \beta_3 + \gamma + 9\pi)/2}\right) \times \left(\frac{i}{\sqrt{2}}\right) \times \left(-\sqrt{2}\left(e^{i\gamma/2}\right)\right) \tag{6.128}$$

$$= (-i)\left(e^{i(\beta_1 + \beta_2 + \beta_3 + 2\gamma + \pi)/2}\right) \tag{6.129}$$

$$= (-i)\left(e^{i\left(\arg z + \arg z' + \beta_2 + \arg z - \arg z' + 2x^+ - 2\arg z + \pi - \beta_2 + \pi\right)/2}\right) \tag{6.130}$$

$$= (-i)\left(e^{i\left(2x^+ + 2\pi\right)/2}\right) \tag{6.131}$$

$$= e^{i(\alpha_1 + \alpha_2 + \pi)/2} \tag{6.132}$$

$\square$

The quaternion $\left(\pi, \frac{x+z}{\sqrt{2}}\right)$, also known as H is the the ZQ equivalent of the Hadamard gate in ZX. This quaternion changes Z rotations to X rotations and vice versa, but is not quite self-inverse, with $H \times H = -1$.

Lemma 6.43. *The quaternion $\left(\pi, \frac{x+z}{\sqrt{2}}\right)$ and its interactions with (α, z) and (α, x):*

$$\left(\pi, \frac{x+z}{\sqrt{2}}\right) \times (\alpha, z) = (\alpha, x) \times \left(\pi, \frac{x+z}{\sqrt{2}}\right) \tag{6.133}$$

$$(\alpha, z) \times \left(\pi, \frac{x+z}{\sqrt{2}}\right) = \left(\pi, \frac{x+z}{\sqrt{2}}\right) \times (\alpha, x) \tag{6.134}$$

$$\left(\pi, \frac{x+z}{\sqrt{2}}\right) \times \left(\pi, \frac{x+z}{\sqrt{2}}\right) = -1 \tag{6.135}$$

The proof of lemma 6.43 is in §C. We now show that the two sequences of rotations in the rule $ZX_{EU'}$ correspond to the same two sequences of rotations in ZQ.

Lemma 6.44. *With the conditions of $ZX_{EU'}$:*

$$(\alpha_1, z) \times H \times (\alpha_2, z) = H \times (\beta_1, z) \times H \times (\beta_2, z) \times H \times (\beta_3, z) \times H \tag{6.136}$$

The proof of lemma 6.44 is in §C. The final lemma before Proposition 6.46 is the evaluation of the parameterised scalar in $ZX_{EU'}$.

Lemma 6.45.

$$ZQ \vdash \begin{array}{c} \boxed{\pi, z} \\ \boxed{H} \\ \boxed{\gamma, z} \end{array} = \lambda_{-\sqrt{2}\left(e^{i\gamma/2}\right)} \tag{6.137}$$

Proof. This is a special case of Lemma 6.35. $\qquad\square$

Proposition 6.46.

$$ZQ \vdash F_Q\left(EU'\right) \tag{6.138}$$

$$i.e. \quad ZQ \vdash \lambda_{e^{\frac{i}{2}(\alpha_1+\alpha_2+\pi)}} \begin{array}{c} \boxed{\alpha_1,z} \\ \bigcirc \\ \boxed{H} \\ \bigcirc \\ \boxed{\alpha_2,z} \end{array} = \lambda_{e^{i(\beta_1+\beta_2+\beta_3+\gamma+9\pi)/2}} \begin{array}{c} \boxed{H} \quad \bigcirc \\ \bigcirc \\ \boxed{\beta_1,z} \\ \boxed{H} \\ \bigcirc \\ \boxed{\beta_2,z} \quad \bigcirc \\ \bigcirc \quad \boxed{0,z} \\ \boxed{H} \quad \boxed{H} \\ \bigcirc \quad \bigcirc \\ \boxed{\beta_3,z} \quad \boxed{\gamma,z} \\ \bigcirc \quad \bigcirc \\ \boxed{H} \end{array} \tag{6.139}$$

Proof.

$$LHS \underset{I_q,Q}{=} \boxed{(\alpha_1,z) \times H \times (\alpha_2,z)} \; \lambda_{e^{i(\alpha_1+\alpha_2+\pi)/2}} \tag{6.140}$$

$$RHS = \lambda_{e^{i(\beta_1+\beta_2+\beta_3+\gamma+9\pi)/2}} \begin{array}{c} \bigcirc \\ \boxed{\pi,z} \\ \bigcirc \\ \boxed{H} \\ \bigcirc \\ \boxed{\gamma,z} \\ \bigcirc \end{array} \boxed{H \times (\beta_1,z) \times H \times (\beta_2,z) \times H \times (\beta_3,z) \times H} \tag{6.141}$$

$$\underset{6.44}{=} \lambda_{e^{i(\beta_1+\beta_2+\beta_3+\gamma+9\pi)/2}} \begin{array}{c} \bigcirc \\ \boxed{\pi,z} \\ \bigcirc \\ \boxed{H} \\ \bigcirc \\ \boxed{\gamma,z} \\ \bigcirc \end{array} \boxed{(\alpha_1,z) \times H \times (\alpha_2,z)} \tag{6.142}$$

$$\underset{6.45}{=} \lambda_{e^{i(\beta_1+\beta_2+\beta_3+\gamma+9\pi)/2}} \; \lambda_{-\sqrt{2}(e^{i\gamma/2})} \boxed{(\alpha_1,z) \times H \times (\alpha_2,z)} \tag{6.143}$$

$$\underset{6.37}{=} \lambda_{e^{i(\beta_1+\beta_2+\beta_3+\gamma+9\pi)/2}} \lambda_{\frac{i}{\sqrt{2}}} \lambda_{-\sqrt{2}(e^{i\gamma/2})} \boxed{(\alpha_1,z) \times H \times (\alpha_2,z)} \tag{6.144}$$

$$\underset{6.42}{=} \lambda_{e^{i(\alpha_1+\alpha_2+\pi)/2}} \boxed{(\alpha_1,z) \times H \times (\alpha_2,z)} \tag{6.145}$$

$\square$

We have shown that for every rule $L = R$ in ZX, $ZQ \vdash F_Q(L) = F_Q(R)$. We have therefore shown that if $ZX \vdash D_1 = D_2$ then $ZQ \vdash F_Q(D_1) = F_Q(D_2)$.

6.4.3   From ZQ to ZX and back again

It remains to be shown that $ZQ \vdash F_Q(F_X(D)) = D$

Proposition 6.47. *Re-translating the Z spider*

$$ZQ \vdash F_Q\left(F_X\left(\;\right)\right) = \qquad \qquad (6.146)$$

Proof.

$$LHS = F_Q\left(\;\right) \underset{6.31}{=} \qquad \qquad (6.147)$$

$\square$

The following lemmas are necessary for the re-translation of the Q node in Proposition 6.52.

Lemma 6.48.

$$ZQ \vdash F_Q\left(\;\right) = \lambda_{1/\sqrt{2}} \qquad \qquad (6.148)$$

Proof.

$$F_Q\left(\;\right) \underset{M}{=} \lambda_{-i} \;\; \underset{M,6.37}{=} \lambda_{1/\sqrt{2}} \qquad \qquad (6.149)$$

$\square$

Lemma 6.49.

$$ZQ \vdash F_Q\left(\;\right) = \lambda_{-\sqrt{2}e^{i\gamma/2}} \qquad \qquad (6.150)$$

Proof.

$$
F_Q\left(\begin{array}{c}\gamma\\ \vdots\\ \pi\end{array}\right) \;\overset{=}{\overset{}{Y}}\;
\begin{array}{c}\text{[}\pi,z\text{]}\\ \vdots\\ \fbox{H}\\ \vdots\\ \text{[}\gamma,z\text{]}\end{array}\;\lambda_{e^{i\gamma/2}}\lambda_i\lambda_i \;\overset{=}{\underset{6.45,\,M}{}}\; \lambda_{\sqrt{2}e^{i\gamma}}
\tag{6.151}
$$

$\square$

Lemma 6.50.

$$
ZQ \vdash F_Q\left(\;\alpha\;\right) = \boxed{\alpha,z}\;\lambda_{e^{i\frac{\alpha}{2}}}
\tag{6.152}
$$

Proof.

$$
LHS = \;\text{(}\bigcirc\text{)} \circ \boxed{\alpha,z} \circ \text{(}\bigcirc\text{)}\;\lambda_{e^{i\frac{\alpha}{2}}} = \boxed{\alpha,z}\;\lambda_{e^{i\frac{\alpha}{2}}}
\tag{6.153}
$$

$\square$

Lemma 6.51.

$$
ZQ \vdash F_Q\left(\;\alpha\;\right) = \boxed{\alpha,x}\;\lambda_{e^{i\frac{\alpha}{2}}}
\tag{6.154}
$$

Proof.

$$
LHS \;\overset{=}{\underset{M}{}}\; \fbox{H} \circ \bigcirc \circ \boxed{\alpha,z} \circ \bigcirc \circ \fbox{H}\;\lambda_{-e^{i\frac{\alpha}{2}}}
\tag{6.155}
$$

$$
\overset{=}{\underset{Q}{}}\; \boxed{H \times (\alpha,z) \times H}\;\lambda_{-e^{i\frac{\alpha}{2}}} \;\overset{=}{\underset{6.43}{}}\; \boxed{H \times H \times (\alpha,x)}\;\lambda_{-e^{i\frac{\alpha}{2}}}
\tag{6.156}
$$

$$
\overset{=}{\underset{6.36,\,N}{}}\; \boxed{\alpha,x}\;\lambda_{e^{i\frac{\alpha}{2}}}
\tag{6.157}
$$

$\square$

Proposition 6.52.

$$ZQ \vdash F_Q \left(F_X \left(\boxed{q} \right) \right) = \boxed{q} \tag{6.158}$$

Where $q = (\alpha_1, z) \times (\alpha_2, x) \times (\alpha_3, z)$, as in Proposition 6.29

Proof.

$$LHS = F_Q \left(\alpha_1 \circ \alpha_2 \circ \alpha_3 \right) \otimes \begin{array}{ccc} \boxed{-\alpha_1/2} & \boxed{-\alpha_2/2} & \boxed{-\alpha_3/2} \\ \pi & \pi & \pi \end{array} \tag{6.159}$$

$$\overset{=}{\underset{6.49, \, 6.48}{}} F_Q \left(\alpha_1 \circ \alpha_2 \circ \alpha_3 \right) \otimes \lambda_{\sqrt{2}e^{-i\alpha_1/2}} \lambda_{\sqrt{2}e^{-i\alpha_2/2}} \lambda_{\sqrt{2}e^{-i\alpha_3/2}} \lambda_{\frac{1}{\sqrt{2}}} \lambda_{\frac{1}{\sqrt{2}}} \lambda_{\frac{1}{\sqrt{2}}} \tag{6.160}$$

$$\overset{=}{\underset{M}{}} F_Q \left(\alpha_1 \circ \alpha_2 \circ \alpha_3 \right) \otimes \lambda_{e^{-\frac{i}{2}(\alpha_1+\alpha_2+\alpha_3)}} \tag{6.161}$$

$$\overset{=}{\underset{6.51, \, 6.50}{}} \left(\boxed{\alpha_1, z} \circ \boxed{\alpha_2, x} \circ \boxed{\alpha_3, z} \right) \otimes \lambda_{e^{-\frac{i}{2}(\alpha_1+\alpha_2+\alpha_3)}} \lambda_{e^{i\alpha_1/2}} \lambda_{e^{i\alpha_2/2}} \lambda_{e^{i\alpha_3/2}} \tag{6.162}$$

$$\overset{=}{\underset{M}{}} \boxed{\alpha_1, z} \circ \boxed{\alpha_2, x} \circ \boxed{\alpha_3, z} \tag{6.163}$$

$$\overset{=}{\underset{Q}{}} \boxed{(\alpha_1, z) \times (\alpha_2, x) \times (\alpha_3, z)} \tag{6.164}$$

$$\square$$

Finally we need the following lemma for Proposition 6.54.

Lemma 6.53.

$$ZQ \vdash F_Q \left(\begin{array}{c} \beta \\ -\beta \end{array} \right) = \lambda_{\sqrt{2}\cos\beta} \tag{6.165}$$

Proof.

$$LHS = \begin{array}{c}\text{[diagram]}\end{array} \lambda_{e^{i\beta_2}} \lambda_{e^{-i\beta_2}} \lambda_i \underset{M}{=} \begin{array}{c}\text{[diagram]}\end{array} \lambda_i \underset{Q}{=} \begin{array}{c}\boxed{(\beta,z) \times H \times (-\beta,z)}\end{array} \lambda_i \qquad (6.166)$$

$$\underset{A}{=} \lambda_{\sqrt{2}\cos\beta} \qquad (6.167)$$

$\square$

Proposition 6.54.

$$ZQ \vdash F_Q\left(F_X\left(\lambda_c\right)\right) = \lambda_c \qquad (6.168)$$

Proof.

$$c = \left(\sqrt{2}\right)^n e^{i\alpha} \cos\beta \quad \text{for some } \alpha, \beta \qquad (6.169)$$

$$LHS = F_Q\left(\begin{array}{c}\text{[diagram]}\end{array} \left(\begin{array}{c}\text{[diagram]}\end{array}\right)^{\otimes n} \begin{array}{c}\text{[diagram]}\end{array}\right) \underset{6.49,\,6.37}{=} \lambda_{\sqrt{2}e^{i\alpha}} \lambda_{\frac{1}{\sqrt{2}}} \lambda_{\frac{1}{\sqrt{2}}} \left(\lambda_{\sqrt{2}}\right)^{\otimes n} F_Q\left(\begin{array}{c}\text{[diagram]}\end{array}\right)$$
$$(6.170)$$

$$\underset{6.53}{=} \lambda_{\sqrt{2}e^{i\alpha/2}} \lambda_{\frac{1}{\sqrt{2}}} \lambda_{\frac{1}{\sqrt{2}}} \left(\lambda_{\sqrt{2}}\right)^{\otimes n} \lambda_{\sqrt{2}\cos\beta} \underset{M}{=} \lambda_{\left(\sqrt{2}\right)^n e^{i\alpha} \cos\beta} = \lambda_c \qquad (6.171)$$

$\square$

We have shown that for each of the generators of ZQ, $ZQ \vdash F_Q\left(F_X\left(G\right)\right) = G$, and since F_Q and F_X are monoidal functors we know that $ZQ \vdash F_Q\left(F_X\left(D\right)\right) = D$ for any diagram D. This concludes our proof of completeness for the rules of ZQ.

6.5   Summary

This chapter introduced ZQ, a complete graphical calculus that extends the phase group concept of ZX. ZQ cannot fit into the categorical framework of Σ Graphical Calculi given in the previous chapter, because the group action of ZQ cannot be expressed as a monoid (and therefore also cannot be expressed as a spider) and is the first qubit graphical calculus with a non-commutative phase group. ZQ allows for the expression of arbitrary single qubit rotations in the form of quaternions, similar to the compiler TriQ.

This expression of arbitrary rotations in ZQ allows for a clearer presentation of the Euler Angle structure than ZX exhibits in the rule (EU'). The author feels this clarity,

and the clarity of the ZQ rules in general, lends ZQ to replacing ZX for pedagogy and research purposes.

In the following chapters we will return to the idea of conjecture synthesis, exploring how conjecture synthesis interacts with our new calculi, and how we can use these results to further improve our inference and verification methods. Future work based on this chapter would include implementing ZQ in a graphical proof assistant, or exploring the category of models of V-GROUP-PROP$_G$. The author also hopes that ZQ will act as a bridge between the optimisation methods of both TriQ and of ZX (such as Refs [Bac+20], [KW20b]).

Chapter 7

Conjecture Inference

In this chapter:

  ▷ We introduce conjecture inference for graphical calculi

  ▷ We provide a link between phase algebras and algebraic geometry

  ▷ We use this geometric link to infer rules from sample instances of those rules

  ▷ We present a simple method for adding !-boxes and phase variables to diagram boundaries

Graphical Conjecture Synthesis[1] produces simple equational theorems, i.e. theorems of the form $L = R$ where L and R are diagrams containing neither !-boxes nor phase variables (see §2.3). In all the calculi we have encountered their complete rulesets can be expressed using either an infinite number of simple rules, or a finite number (around 20) of decorated rules. We therefore would like to know the answer to 'how can we find rules containing !-boxes and phase variables?' We break this down into two situations:

  • Finding these rules with certainty, and

  • Finding these results in a way we can easily verify (e.g. by checking a few more examples).

This chapter covers the inference aspect of this process, with the next chapter (§8) devoted to the task of verification. Our aim in conjecture inference is given in the definition below, but we note that it is simply a small part of the greater philosophical program called 'The Problem of Induction'. This Problem began with David Hume [Hum03, §1.iii.6] and concerns the construction of new ideas from existing evidence.

Definition 7.1 (Conjecture Inference)**.** Conjecture Inference is the algorithmic act of hypothesising families of theorems, given a few examples from that family.

---

[1]Outside of the contributions of this thesis. See §3.

In this chapter we also demonstrate a link between phase algebras and Algebraic Geometry (§7.2). This link is novel, useful, and neatly generalises existing concepts in the literature. We will use this link to clarify the relationship between rules with and without side conditions, explore concepts like rebasing of variables, and then rely on geometric ideas as a core part of our inference process. This geometric framework builds on the earlier ideas of phase homomorphism pairs (§5), where these pairs will be rephrased as symmetries of the geometric space. This idea will also be carried forward when we come to discuss verification: The geometric interpolation discussed in this chapter is directly related to the interpolation discussed in the next, and we combine these ideas with the above symmetries (and some Galois Theory) for a pleasantly surprising result in §8.3.

7.1   Adding parameters at boundaries

There is a very simple way to hypothesise adding !-boxes and phase variables to certain boundary spiders in ZX, ZH, ZW, and ZQ rules, and what's more the method comes with a proof of correctness, meaning that we don't need to verify any more examples. For each of our calculi we can use spider laws to infer more general results from simple equations. These results are included in part for posterity (any proof assistant that uses these calculi should ideally be capable of automatically inferring variations on Proposition 7.2) but also as a contrast to the methods of §7.2. The hypotheses inferred in §7.2 do not come with proofs of correctness and must be verified in some manner (the topic of §8).

In order to signify that this process can be applied at the boundary of a diagram we use the following to signify an arbitrary diagram D:

$$(7.1)$$

Proposition 7.2. *A ZX_G equation of the form*

$$b, c \text{ fixed phases} \in G$$
$$\text{non-zero, fixed number of inputs} \qquad (7.2)$$
$$D, D' \text{ arbitrary diagrams}$$

implies and is implied by the family of ZX_G equations parameterised over the phase

variable α and a !-box:

$$
\left\{
\begin{array}{c}
\cdots \\
\boxed{D} \\
\cdots \\
\boxed{\alpha} \\
\boxed{}
\end{array}
=
\begin{array}{c}
\cdots \\
\boxed{D'} \\
\cdots \\
\boxed{\alpha + c - b} \\
\boxed{}
\end{array}
\right\}_{\alpha,!}
\qquad
\begin{array}{c}
\alpha \text{ phase variable } \in G \\
\text{!-box on inputs}
\end{array}
\qquad (7.3)
$$

This holds for Z and for X spiders, providing the spider is the same colour on both sides of the rule.

Proof. We can apply a spider s with phase $(\alpha - b)$ to both sides of the top rule r. The spider s has as many outputs as r has inputs, and can have any number of inputs. For the converse we specify the !-box to have has many instances as r has inputs, and we instantiate α to the value b. $\qquad\square$

Remark 7.3. Proposition 7.2 and its proof both generalise easily to the spiders of ZW. For ZQ one can generalise either the phase of a Q node, or add a !-box to a Z-spider, depending on which is joined to the boundary. In RING_R one can add a !-box to the inputs or outputs of the multiplication spider, or to the inputs of the addition spider. In both cases you can also generalise the phase. The Z-spider of ZH has a phase-free analogue of Proposition 7.2, however the H-box result requires a little more work and is presented after Lemma 7.4.

Lemma 7.4 (n-joined ZH spider law)**.** *In the ZH calculus:*

$$
\begin{array}{c}
\cdots \\
\boxed{a} \\
\boxed{} \overset{n}{\cdots} \boxed{} \\
\boxed{b} \\
\cdots
\end{array}
=
\boxed{1 + (\tfrac{-1}{2})^n (b-1)(a-1)}
\quad
\boxed{2^n}
\qquad
\begin{array}{c}
a, b \in \mathbb{C} \setminus \{1\} \\
n > 0
\end{array}
\qquad (7.4)
$$

Since the proof isn't relevant to the discussion of conjecture inference we defer it to §D.

Proposition 7.5. *A simple ZH equation of the form:*

$$
\begin{array}{c}
\cdots \\
\boxed{D} \\
\cdots \\
\boxed{b} \\
\overset{n}{\cdots}
\end{array}
=
\begin{array}{c}
\cdots \\
\boxed{D'} \\
\cdots \\
\boxed{c} \\
\overset{n}{\cdots}
\end{array}
\qquad
\begin{array}{c}
b, c \text{ fixed phases } \in \mathbb{C} \setminus \{1\} \\
n, \text{ non-zero number of inputs} \\
D, D' \text{ arbitrary diagrams}
\end{array}
\qquad (7.5)
$$

implies and is implied by the family of equations parameterised over α and a !-box:

$$\left\{ \vcenter{\hbox{}} \right\}_{\alpha,!} \qquad \begin{array}{c} \alpha \in \mathbb{C} \\ \textit{!-box on inputs} \end{array} \qquad (7.6)$$

Proof. Apply the following diagram to the bottom of each side of Equation (7.5):

$$(7.7)$$

Then apply lemma 7.4. As with the ZX case we justify the !-box by saying that the above is true for any number of inputs and value of α. For the converse we specify the !-box to have has many instances as needed, and we set α to b. $\square$

7.2   Linear rules, parameter spaces, and algebraic geometry

We move now to a method of conjecture inference that does not come with a proof of correctness. The fundamental idea behind this method is to transform the problem of conjecture inference into a problem in algebraic geometry. Rather than seeing a parameterised family of diagrams as a *set* of diagrams we see them as points defining a surface in some affine space. The question of conjecture inference then becomes 'given a sample of these points, can I reconstruct the surface?' In order to frame this question correctly we begin with some necessary definitions (§7.2.1), show how to view parameterised families as submodules (§7.2.2), show how to get back to parameterised equations from submodules (§7.2.3), and finally show how to find submodules from sample points (§7.2.4).

One way of thinking about the content of this section is that in the usual presentation of our calculi and rules we have as much information as possible in the algebra written as phases, and as little information as possible in the side conditions. This section describes how to move from the algebraic paradigm into a geometric one; describing everything as restrictions on a space (i.e. as side conditions). In the case of linear rules over $\mathbb{Z}$-modules we show how to go in the other direction as well (from geometry to algebra), describing the linear geometric constraints using just the algebra in the phases. This concept neatly generalises the idea of linear rules for ZX already present in the literature.

7.2.1   Skeletons and parameter spaces

We will assume for the rest of this chapter that the phase algebra we are considering is R-linear for some commutative ring R. In the case of ZX we would consider the phase group as a $\mathbb{Z}$-module, for RING_R, ZH_R and ZW_R we would consider the phase ring as an R-module. ZQ can be considered as an $\mathbb{R}$-module, where a phase is an element of $\mathbb{R}^4$, but only some of the points in the space result in valid phases.

The key to this section is the novel separation of a diagram into its skeleton and parameter space. The skeleton holds all the graph information except for the phases, and the parameter space holds the information from the phase algebra.

Definition 7.6 (Skeleton)**.** The *skeleton* of a diagram, written $|D|$, is the phase-free version of that diagram. The skeleton of a rule is the pair of skeletons from its two diagrams.

Example 7.7 (ZX diagram skeleton)*.* An example diagram and skeleton:

$$D \quad \text{} \qquad\qquad |D| \quad \text{} \qquad\qquad (7.8)$$

Definition 7.8 (Parameter space)**.** The *parameter space* of a skeleton is the direct product (in R-mod) of all the phase algebras of the nodes in the skeleton.

$$P := \prod_{\text{nodes in the skeleton}} \{\text{Phase algebra of that node}\} \qquad\qquad (7.9)$$

The parameter space of a rule is the direct product of the parameter spaces of each diagram.

Remark 7.9*.* Some nodes contribute trivially to the parameter space. For example the Z spider in ZH is always blank. We treat this as the trivial R-module, $\{0\}$, which contributes nothing to the overall product P. As such these cases will be ignored from here on.

Remark 7.10*.* In the case of calculi such as ZQ, where the phase algebra is a four-dimensional $\mathbb{R}$-module, the parameter space can have many more dimensions than the rule has vertices. In later subsections of this chapter this may require viewing the phase of a Q node as four distinct, real numbers rather than as a single quaternion, but the alterations that need to be made to the methods are small and will not be mentioned again. In essence they are 'treat this node labelled by an element from R^n as n nodes each labelled by an element of R'.

Example 7.11 (Clifford+T ZX parameter space)*.* Consider, as an example, the left hand side of a simplified Clifford+T ZX spider law:
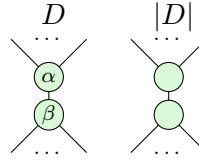
This skeleton has parameter space:

$$< \pi/4 > \times < \pi/4 > \quad \simeq \quad \mathbb{Z}_8 \times \mathbb{Z}_8 \qquad (7.10)$$

Which is the product of the two cyclic phase groups, viewed as $\mathbb{Z}$-modules, one for each node. Assuming we have made (or retained) a choice of which phase group corresponds to which node then every point in the parameter space corresponds to a diagram. For example the point $p = (\pi/2, 3\pi/4)$ in the parameter space corresponds to the diagram:

$$(7.11)$$
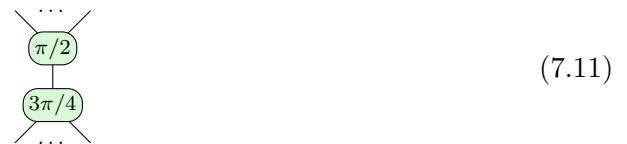
Definition 7.12 (Sound points in a parameter space)**.** A point p in a parameter space is *sound* if the corresponding rule, with phases as described by p, is sound (see Definition 2.27). Any point in P that does not correspond to a sound rule is *unsound*.

7.2.2 Equation families as submodules

In this subsection we view a family of equations parameterised by phase variables as a collection of simple rules that all have the same skeleton, and a collection of sound points in the parameter space. This will allow us to rephrase families parameterised by phase variables as submodules of a parameter space.

Example 7.13 (Parameter space of the spider rule). Here is a (slightly unusual) way of presenting the simplified spider rule in ZX_G:

$$(\alpha, \beta, \gamma) \in P$$
$$\gamma = \alpha + \beta \qquad (7.12)$$

We have given the parameterisation as a collection of points in the parameter space P, subject to the condition $\alpha + \beta = \gamma$.

Remark 7.14*.* This presentation of rules is not uncommon; see for example the *EU* rule of [Vil19, §2.3].

We will now introduce the bare minimum of algebraic geometry; the notion of a *variety* or *zero set*. It is this tool that links the two notions of algebra (in the form of a system of equations) and geometry (in the form of a topological shape).

Definition 7.15 (Affine algebraic variety [Smi+04, §1.1]). A complex *affine algebraic variety* is the common zero set of a collection $\{F_i\}_{i \in I}$ of complex polynomials on complex n-space $\mathbb{C}^n$.

Definition 7.16 (Linear rule). A parameterised family of rules is *linear* if the sound points in P are the zero set of a collection of linear polynomials on P.

Remark 7.17 (Parameter space restrictions as side conditions). For a calculus like ZQ, where only certain points in the parameter space correspond to valid diagrams, we impose these same restrictions when talking about rules. I.e. we say that a parameterised family of rules is linear if the sound and valid points in P are the intersection of the zero set of a collection of linear polynomials on P with the valid points in P. We could even define ZQ in such a way that these 'valid points in P' were phrased as side conditions on the generators and rules.

This definition is, on the surface, different to the definition of a linear rule given in [JPV18b], which we reproduce (in language consistent with the rest of this thesis) as:

Definition 7.18 (Previous definition of linear rule [JPV18b, Definition 1]). A ZX-diagram is linear in $\alpha_1, \ldots, \alpha_k$ with constants in $C \subset \mathbb{R}$, if it is generated by Z-spiders with phase E, X-spiders with phase E, the Hadamard node, and compact closed wire structure, where E is of the form $\sum_i n_i \alpha_i + c$ with $n_i \in \mathbb{Z}$ and $c \in C$.

Thankfully we can show these definitions coincide, starting with:

Proposition 7.19. *Every linear ZX rule in the sense of Definition 7.18 is a linear rule in the sense of Definition 7.16.*

Proof. Give every spider in the linear rule an index $j \in J$, which as per Definition 7.18 has phase $\sum_i n_{i,j} \alpha_i + c_j$. Assign to each spider the parameter β_j, and construct the system of linear equations (as in Definition 7.16) as the set $\{\beta_j = \sum_i n_{i,j} \alpha_i + c_j\}_{j \in J}$ □

Remark 7.20. The converse to Proposition 7.19 is given as Corollary 7.25 later on in this chapter. While this shows that the two definitions are equivalent in the case of ZX_G, we prefer the geometric definition because it can be generalised to other calculi; for example it is unclear how to generalise Definition 7.18 to the case of ZQ.

Example 7.21 (Linear Clifford+T ZX rule). The following Clifford+T ZX rule is linear in the sense of Definition 7.16:

$$\{\alpha = 0, \delta = 0, \beta - \gamma = 0\} \qquad (7.13)$$

The affine, linear $\mathbb{Z}$-submodule described by the linear polynomials is $\mathbb{Z}(0, \pi/4, \pi/4, 0)$.

Example 7.22 (Rule with multiple linear subspaces). The following Clifford+T ZX rule (although very similar to Example 7.21) is not linear, but the union of four different

sound (up to scalar), linear rules on the same skeleton:

$$\{\alpha = 0, \delta = 0, \beta - \gamma = 0\}$$
$$\{\beta = 0, \gamma = 0, \alpha - \delta = 0\}$$
$$\{\alpha - \pi = 0, \delta - \pi = 0, \beta + \gamma = 0\}$$
$$\{\beta - \pi = 0, \gamma - \pi = 0, \alpha + \delta = 0\}$$

(7.14)

This is because it is sound on the following linear varieties of P:

defining polynomials	affine submodule	
$\alpha = 0, \delta = 0, \beta = \gamma$	$(0,0,0,0) + \mathbb{Z}(0, \pi/4, \pi/4, 0)$	(7.15)
$\beta = 0, \gamma = 0, \alpha = \delta$	$(0,0,0,0) + \mathbb{Z}(\pi/4, 0, 0, \pi/4)$	(7.16)
$\alpha = \pi, \delta = \pi, \beta = -\gamma$	$(\pi,0,0,\pi) + \mathbb{Z}(0, \pi/4, -\pi/4, 0)$	(7.17)
$\beta = \pi, \gamma = \pi, \alpha = -\delta$	$(0,\pi,\pi,0) + \mathbb{Z}(\pi/4, 0, 0, -\pi/4)$	(7.18)

For example the rule given by this skeleton and just the variety given by (7.15) is Example 7.21.

Remark 7.23. The rule in Example 7.22 can be seen as applications of the spider and π-commutation rules, and their colour-swapped counterparts, (see (S2) and (K) of Ref. [JPV18b]) representing the four linear subspaces we have exhibited.

Having shown how to view families of equations as submodules we now work backwards; from submodules to families of equations. The benefit of the submodule viewpoint is deferred until §7.2.4, where we examine how to hypothesise larger submodules containing a known collection of points.

7.2.3   Submodules as families of equations

Assume we have been given a sound, affine submodule $q + Q$ of P. In the case where Q is finitely generated, and R admits an Euclidean function, we can find the Hermite Normal Form to simplify relationships between generators. (In the case where R is a field we can do Gaussian Elimination, and the following is even simpler.)

Proposition 7.24. *It is possible to describe any affine, linear R-submodule of a parameter space as a single diagram with phases that are linear in some set of phase variables. Additionally we can ensure the set of phase variables is of minimal cardinality.*

Proof. We provide here an algorithm that assigns to each vertex such a phase. The input for the algorithm is the rule skeleton, and a description of the finitely generated, linear R-submodule $q + Q$.

We treat Q as a matrix where each row corresponds to a generator, and each column corresponds to a vertex in the rule. We then find a linearly independent set of generators

for the space Q by putting the matrix Q into Hermite Normal Form. I.e. we find matrices U and H such that $H = UQ$, where U is a product of elementary column and row operations, and H has the following properties:

- The leading coefficient of each row is not in the same column as the leading coefficient of any other row,

- The leading coefficient of each row has only the entry 0 in that column in any of the rows below it,

- If the leading coefficient of a row has value r then the values in places above it have smaller Euclidean degree than r,

- All rows containing only 0 come after any rows that are not all 0,

- All non-zero rows are linearly independent of the other non-zero rows.

This is a standard procedure in mathematics software (see, for example, an implementation in Ref. [BCP97]), and all of these properties can be shown to be obtainable via Euclid's Algorithm. The matrix H corresponds to a linearly independent set of generators of Q, i.e. if H is split into rows $\{r_j\}_{j=1}^n$ then $Q = Rr_1 + Rr_2 + \cdots + Rr_n$.

To show that all linear independent sets of generators for Q are the same size: Given two sets of generators $B := \{b_j\}$ and $B' := \{b_k'\}$, each linearly independent, and with $|B| < |B'|$, one can construct the matrix C where the kth row is $\{r_j\}_{j=1}^{|B|}$ and $\sum_j r_j b_j = b_k'$. If we find the Hermite Normal Form of C the result is upper triangular of shape $|B| \times |B'|$, so the final row must be all 0s, and so there must be a linear dependence inside B', which is a contradiction. Therefore any linear independent sets of generators for Q are of the same size.

We translate the matrix H to a diagram by assigning to each non-zero row of H a phase variable α_k, and for each vertex v_j in the skeleton (with corresponding column $\{c_l\}_{l=1}^m$) that vertex is assigned the phase $q_j + \sum_l c_l \alpha_l$, recalling that q was the affine shift of the original linear subspace. □

Corollary 7.25. *The two definitions of a linear rule (Definition 7.16 and Definition 7.18) coincide. One direction is Proposition 7.19, the other is Proposition 7.24.*

Example 7.26 (Two different subspace representations as rules). The skeleton of the rule S1 for $ZX_{\pi/4}$ has two vertices on the left hand side, and one on the right, which we label v_1, v_2 and v_3

$$\tag{7.19}$$

P is therefore isomorphic to $\mathbb{Z}_8 \times \mathbb{Z}_8 \times \mathbb{Z}_8$, and we will assume for this example that we already know that the rule is sound for $p \in q + Q$ where:

$$q := (0, 0, 0) \tag{7.20}$$
$$Q := \mathbb{Z}(1, 0, 1) + \mathbb{Z}(0, 1, 1) \tag{7.21}$$

As in the proof of Proposition 7.24 we consider the Q part as a matrix:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{7.22}$$

Which is already in Hermite Normal Form. We assign to each row a phase variable, and then add together every entry in each column:

$$\begin{bmatrix} \alpha & 0 & \alpha \\ 0 & \beta & \beta \end{bmatrix} \qquad (\alpha, \beta, \alpha + \beta) \tag{7.23}$$

This gives us the phases we assign to the skeleton of the rule:

$$v_1 \mapsto \alpha \qquad\qquad v_2 \mapsto \beta \qquad\qquad v_3 \mapsto \alpha + \beta \qquad\qquad$$ $$\tag{7.24}$$

Suppose instead we started with the following presentation of the same linear submodule:

$$q := 0 \tag{7.25}$$
$$Q' := \mathbb{Z}(-1, 1, 0) + \mathbb{Z}(0, 1, 1) \cong Q \tag{7.26}$$

Corresponding to the matrix:

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{7.27}$$

If we applied the same rewriting to get a family of equations (without finding Hermite Normal Form) then we would get:

$$\begin{bmatrix} -\alpha & \alpha & 0 \\ 0 & \beta & \beta \end{bmatrix} \qquad (-\alpha, \alpha + \beta, \beta) \tag{7.28}$$

This gives us the 'same' rule:

$$v_1 \mapsto -\alpha \qquad\qquad v_2 \mapsto \alpha + \beta \qquad\qquad v_3 \mapsto \beta \qquad\qquad \tag{7.29}$$

Equations (7.24) and (7.29) are the same rule, in that they describe the same set of simple rules, but they are presented differently because they reflect the different ways of presenting the underlying affine, linear spaces Q and Q'.

Remark 7.27. This use of the Hermite Normal Form is not unexpected, since this process bears a strong resemblance to the way the matcher works in the software Quantomatic [KZ15]. The matcher is the subroutine that matches instances of rules onto diagrams, and in doing so needs to match variable names onto concrete values.

Definition 7.28 (Independent phase variables)**.** We call the phase variable α in some rule *independent* if there is any vertex with the phase $\alpha + c$, where c is a non-variable element of the phase algebra.

Remark 7.29. This 'rebasing' of the affine linear subspace actually allows one to shift the independent phases around in a rule; in particular from one side to the other. Doing so in a proof assistant would allow for clearer representations of rule inverses, where the independent variables are presented on the left hand side of the rule. This can be achieved by choosing the order of the vertices in the method of Proposition 7.24 such that the vertices from the left hand side of the rule appear to the left (as columns in the matrix Q) of the vertices from the right hand side of the rule.

Lemma 7.30. *If the phase algebra is a K-module for some field K then all phase variables can be re-based as independent phase variables.*

Proof. Working over a field K we can put the matrix Q into reduced row echelon form via Gaussian Elimination, i.e. each leading coefficient is 1 and is the only entry in that column. When translated into phases this results in a phase of $\alpha + c$, where c is a non-variable element of the phase algebra. $\square$

Remark 7.31. We mentioned in §3.2 that it seems unlikely that it is possible to find a ruleset of Universal ZX that can be expressed using linear rules and the group action. This was the impetus for ZQ, and the author notes that the rules of ZQ presented in §6.2.1 use only linear rules (as an $\mathbb{R}$-algebra) and the group action.

7.2.4   Finding submodules

Now that we know how to represent linear rules as affine linear submodules, we want to go about the business of inferring which submodules lead to sound rules. Given a skeleton rule $|D| = |D'|$, we know it is sound for some (possibly empty) subset of the parameter space P. The question is which linear affine submodules of P to check for soundness. We suggest here three ways of hypothesising larger linear submodules containing the points we already know to be sound in P:

- **Full Linear Interpolation.** Given two points p and q sound in P: Extend the points into a line $p + R(q - p)$.

- **Sparse Linear Interpolation.** Given two points p and q sound in P, extend the points into a sparse line $p + \mathbb{Z}r$, where there is some $a \in \mathbb{Z}$ such that $a \times r = q - p$. Note that this line does not necessarily lie inside the parameter space, and in such cases this technique cannot be applied.

- **Summation.** Given two, affine submodules $s + S$, $t + T$, sound in P, that intersect at q: The affine submodules can be re-based to $q + S + (s - q) = q + S$ and $q + T + (t - q) = q + T$. Consider $q + S + T$.

Remark 7.32 (Higher Order Interpolation). With the exception of the (EU) rule (and variants) all of the rules for all of the calculi discussed in this thesis are linear, and the above interpolation methods will only infer linear subspaces. We can, however, still apply higher order methods, see Ref. [GS00] for a survey of higher-order polynomial interpolation techniques. These higher-order techniques would still not be able to infer the (EU) rule of ZX since it does not express a polynomial relationship.

Remark 7.33. This technique is similar to the method of altering the existential and universal quantifiers in term-based conjecture synthesis, mentioned in Example 3.5. Our method makes use of both the linear structure and the information from multiple theorems (sound points) at once.

Example 7.34 (Finding the Clifford+T ZX spider law's subspace). Suppose, using $\text{ZX}_{\pi/4}$, that you know the spider law of (7.12) holds true for $p = (0, 0, 0)$, $q = (\pi/2, 0, \pi/2)$.

- Sparse Linear Interpolation suggests the subspace $Q_1 := (0, 0, 0) + \mathbb{Z}(\pi/2, 0, \pi/2)$

- Full Linear Interpolation suggests the subspace $Q_2 := (0, 0, 0) + \mathbb{Z}(\pi/4, 0, \pi/4)$

- Suppose we also knew the subspaces Q_2 and $Q_3 := (0, 0, 0) + \mathbb{Z}(0, \pi/4, \pi/4)$ were sound, then summation would suggest the subspace $Q_4 := (0, 0, 0) + \mathbb{Z}(\pi/4, 0, \pi/4) + \mathbb{Z}(0, \pi/4, \pi/4)$, which would give you every sound point in P.

What these methods require is some non-exhaustive method of verifying the inferred subspaces of P. The chapter on verification (§8) answers this problem for phase variables, and is inspired by this chapter's link to algebraic geometry, in particular by the geometric notion of degree.

Remark 7.35. We have been working with linear varieties to satisfy the constraints of ZX and ZQ. When working with phase rings we can consider non-linear varieties (i.e. higher degree polynomials in the phase variables) although we have not developed heuristics for which non-linear polynomials to conjecture.

7.2.5   Phase homomorphisms and parameter space symmetries

The concepts of this chapter can also be combined with that of §5.3, where we constructed the phase homomorphism pair $(\phi, \tilde{\phi})$, where $\phi : R \to R$ (lifted to $\hat{\phi}$ acting on the diagrams of RING_R) and $\tilde{\phi}$ acting on matrices in R-bit.

Proposition 7.36. *Given a* RING_R *equation* $D_1 = D_2$ *with parameter space* P*, then if* $(p_1, \ldots, p_n) \in P$ *is sound then* $(\phi(p_1), \ldots, \phi(p_n))$ *is also sound for any ring endomorphism* ϕ*.*

Proof. The action of $(p_1, \ldots, p_n) \mapsto (\phi(p_1), \ldots, \phi(p_n))$ is precisely the action of $\hat{\phi}$ in Proposition 5.40. □

This shows that any phase homomorphism pair is a symmetry of the parameter space. There is one final piece to the conjecture inference puzzle, and that is conjectures involving !-boxes. In order to approach this problem we first need to know about a property called the *join* (Definition 8.41). The join will tell us how far into a series of !-box expansions we need to check before the pattern continues indefinitely. This line of thought is continued in Remark 8.59, after the results on finite verification.

7.2.6   Varying the skeleton

§7.2 has so far concerned itself with investigating the parameter space of an equation with a fixed skeleton. We will now explore what happens when we vary a skeleton, initially by adding components via ∘ or ⊗.

Proposition 7.37. *Given an equation* $D = D'$ *with parameter space* P*, a sound subspace* Q *of* P *embeds as the sound subspace* $Q \oplus R \subset P \oplus R$ *for the equation* $D \otimes E = D' \otimes E$ *and for the equation* $D \circ E = D' \circ E$*.*

Proof. Write M_q for the interpretation of the diagram D at point $q \in Q$. Likewise write M_q' for the interpretation of D' at point q, and N_r for the interpretation of E at the point r in R (the parameter space of the diagram E). Since q is a sound point in Q:

$$M_q = M_q' \tag{7.30}$$

$$\therefore \quad M_q \circ N_r = M_q' \circ N_r \quad \forall r \tag{7.31}$$

$$\therefore \quad M_q \otimes N_r = M_q' \otimes N_r \quad \forall r \tag{7.32}$$

Therefore the sound subspace $Q \subset P$ embeds as $Q \oplus R \subset P \oplus R$. □

What if, rather than compose diagrams, we want to change the underlying graph of existing ones? We saw earlier in this chapter how to generalise from a single boundary wire to a boundary wire with a !-box, and here we shall go from no boundary wire to a single boundary wire. In this next proposition we will assume the diagrams are ZX_G diagrams but the result is easy enough to generalise to diagrams in other graphical calculi.

Proposition 7.38. *Starting with an equation $D = D'$ of ZX_G diagrams with parameter space P: Let e_α be the vector in P that is 0 everywhere except for a 1 at position α, and let α and β be labels for two vertices of the same colour in D and D' respectively. If, and only if, $Q + \mathbb{Z}(e_\alpha + e_\beta)$ is sound in P then the diagram equation formed by adding an additional boundary connected to α and β on each side respectively is sound at $Q + \mathbb{Z}(e_\alpha + e_\beta)$.*

Proof. We begin by picking a point $q \in Q$, and then rewriting the diagrams D and D' as compositions of the vertices α and β with the subdiagrams $\bar{D}$ and $\bar{D}'$. We are assuming both α and β label Z vertices, the proof for X vertices is identical, just with colours swapped.

$$[\![\, D_q \,]\!] = [\![\, D'_q \,]\!] \qquad\qquad \text{} \qquad\qquad (7.33)$$

We then use the spider law to extend α and β downwards:

$$\text{} \qquad\qquad (7.34)$$

The above equation is sound for all $q \in Q$. Since $Q + \mathbb{Z}(e_\alpha + e_\beta)$ is sound in P we therefore know that $q + \pi(e_\alpha + e_\beta)$ is sound in P. Accordingly the following equation is also sound:

$$\text{} \qquad\qquad (7.35)$$

We therefore know that the following diagram is sound for all values of q, since it is

sound on a basis of inputs:

$$
\begin{array}{ccc}
\boxed{\bar{D}_q} & & \boxed{\bar{D}'_q} \\
q_\alpha & = & q_\beta
\end{array}
\tag{7.36}
$$

By Proposition 7.2 the following diagram is sound for all values of q, γ being a phase variable:

$$
\begin{array}{ccc}
\boxed{\bar{D}_q} & & \boxed{\bar{D}'_q} \\
q_\alpha + \gamma & = & q_\beta + \gamma
\end{array}
\tag{7.37}
$$

Therefore $q + \mathbb{Z}(e_\alpha + e_\beta)$ is sound for all values of Q.

For the converse, with γ a phase variable not appearing in D or D':

$$
\begin{array}{ccc}
\boxed{\bar{D}_q} & & \boxed{\bar{D}'_q} \\
q_\alpha + \gamma & = & q_\beta + \gamma
\end{array}
\qquad \text{sound at } q
\tag{7.38}
$$

$$
\Longrightarrow
\begin{array}{ccc}
\boxed{\bar{D}_q} & & \boxed{\bar{D}'_q} \\
q_\alpha + \gamma & = & q_\beta + \gamma \\
\circ & & \circ
\end{array}
\qquad \text{sound at } q
\tag{7.39}
$$

$$
\Longrightarrow
\begin{array}{ccc}
\boxed{\bar{D}_q} & & \boxed{\bar{D}'_q} \\
q_\alpha + \gamma & = & q_\beta + \gamma
\end{array}
\qquad \text{sound at } q
\tag{7.40}
$$

$\square$

We showed earlier on in this chapter that parameter subspaces can be reflected in the phase algebra. In the case of Proposition 7.38 we saw the existence of a specific sound subspace, $Q + \mathbb{Z}(e_\alpha + e_\beta)$, and deduced the ability to add boundaries to the diagram. In terms of phases this sound subspace would appear, given the right choice of presentation, as a variable appearing exactly once in D and once in D', as in Equation (7.40).

Remark 7.39 (From arbitrary many to one to none)*.* In Proposition 7.2 we showed that if a !-boxed boundary is connected to the same colour vertex on each side of the equation then we only need to check the case where the !-box is expanded once. In Proposition 7.38

we showed that if we are already calculating the sound subspaces then we only need to check the case where the !-box is expanded 0 times; i.e. without any boundaries at all. From the soundness of the equation with 0 boundaries, Proposition 7.38 will let us deduce the soundness of the equation with 1 boundary, and then Proposition 7.2 will let us generalise immediately to a !-boxed version.

Remark 7.40 (From sparse to full). Readers may also have noticed that we did not use the full set $Q + \mathbb{Z}(e_\alpha + e_\beta)$, simply two points in it, q and $q + \pi(e_\alpha + e_\beta)$. We will see in the next chapter how the existence of two sound, distinct points inside a subspace known to be linear implies the entire subspace is sound. The proposition could have been phrased: "If q and $q + a(e_\alpha + e_\beta)$ are distinct and sound in P then $q + \mathbb{Z}(e_\alpha + e_\beta)$ is sound for the diagram equation formed by adding an additional boundary connected to α and β on each side respectively." The current phrasing emphasises the three-way link between the linear subspace $\mathbb{Z}(e_\alpha + e_\beta)$, the presentation of some variable γ appearing only on vertices α and β, and the ability to add a boundary to the vertices.

7.3   Summary

This chapter took the existing notion of linear rules in ZX and generalised it to apply to all our graphical calculi, and also to beyond linear rules. The purpose of this was to find ways to generalise theorems from just a few examples, but it also shows how to link the two different presentations of rules (as phase algebras or as side conditions) and unites the algebraic presentation of diagrams with a geometric notion of spaces. With this framework in place we gave heuristics for inferring new polynomial hypotheses based on an existing sample of theorems. We also examined generalisations at boundaries; ways to deductively add phase variables and !-boxes to theorems, immediately increasing their applicability. Even without reference to conjecture synthesis any proof assistant benefits from increasing the applicability of lemmas, and in doing so reducing the number of proof steps for the user.

Our examination of the parameter space showed that sound subspaces were preserved under ∘ and ⊗, that examination of the spaces could lead to important generalisations (such as adding boundaries to diagrams), and that phase homomorphisms led to symmetries of sound subspaces. We will combine this knowledge of symmetries with geometric constraints on the sound subspaces in the next chapter, but already we have an indication that the study of these subspaces could be as rich a topic as the study of quantum graphical calculi. Parameter spaces give a novel and important method of studying parameterised diagram equations.

Future work includes further examination of parameter spaces; in particular their symmetries, other generalisations like Proposition 7.38, and the effects of common operations on the skeleton such as adding or removing an edge. Further work should also be done on deductive generalisations, as any such theorem immediately improves conjecture synthesis runs. Our focus will now turn to the problem of verifying these inferred

hypotheses.

Chapter 8

Conjecture Verification

In this chapter:

> ▷ We introduce conjecture verification for phase ring and phase group calculi

> ▷ We demonstrate the ability to finitely verify certain classes of hypotheses

> ▷ We discuss the relationship between conjecture verification and conjecture inference

> ▷ We use phase homomorphisms to simplify the process even further over qubits

Conjecture Verification is the act of checking whether a hypothesis is true. In this chapter we will only consider hypotheses of the form $[\![\, D_1 \,]\!] = [\![\, D_2 \,]\!]$, i.e. semantic equality of two diagrams. This is in contrast to the syntactic question of whether $\mathcal{R} \vdash D_1 = D_2$ for some ruleset $\mathcal{R}$. We only consider the question of semantic equality because our ultimate aim is Conjecture Synthesis (§3), in which we try to construct a ruleset that is sound and useful, rather than starting with a ruleset and showing it is complete.

This is the final results chapter of this thesis, providing verification methods for the inference procedures discussed in §7 (among others). In particular the verification of phase variables in §8.2 relies on ideas of polynomial fitting that mirror those of §7.2. §8.3 then builds on this foundation by including results from Galois Theory and the phase homomorphism pairs of §5.

These verification methods rely on the structure inherent in phase algebras, and on a novel presentation of !-box expansions that allow us to view these primarily as ∘ rather than ⊗ products.

8.1 Motivation

In earlier chapters, with the exception of parts of §5, we considered a phase algebra to be a founding property of a calculus; without specifying a phase algebra we could not

specify the generators and so could not construct diagrams. We present here a different point of view: Consider a 'base' calculus with a phase algebra $\mathcal{A}$ and without !-boxes. We call diagrams constructed in this calculus *simple* (Definition 2.25). By contrast we then consider a 'higher' calculus, identical to the base calculus except that we now allow formal variables in $\mathcal{A}$ (Definition 2.21), and also !-boxes (Definition 2.23).

Our motivation for this perspective is that this is how many papers present their rulesets. For example the calculus $\mathrm{ZH}_\mathbb{C}$ formally only allows elements of the complex numbers as labels on H-boxes, but the rules presented in [BK19] use !-boxes and formal variables, and the meaning is clear from the context. Our ultimate aim is for computers to be able to reason with a similar degree of expressive power; to be able to hypothesise and then verify equations between decorated diagrams, initially assuming only that they are capable of reasoning with simple diagrams.

Using $\mathrm{ZH}_\mathbb{C}$ as an example we list in Figure 8.1 some structures of interest: The sets of diagrams that fit certain criteria, as well as their interpretations as matrices or sequences of matrices. The *ev* (evaluation) and *as* (assignment) maps generalise in the obvious ways. From these maps, and ones very similar to them, we can construct the commutative cube in Figure 8.2. Our reason for doing so is that the map of interest $(\llbracket \cdot \rrbracket : \mathrm{ZH}^!_{\mathbb{C}[X]} \to (\mathbb{N} \to \mathrm{Mat}_\mathbb{C}))$ which interprets the 'higher' structure can then be linked to the map we assume we have some knowledge of $(\llbracket \cdot \rrbracket : \mathrm{ZH}_\mathbb{C} \to \mathrm{Mat}_\mathbb{C})$ which interprets the 'base' structure. Although we will use the language of category theory (such as commutative diagrams) in this section the structures of interest are considered as discrete categories.

The property we are interested in for this chapter is that of finite sampling: Determining the nature of a function by knowing its value at enough places. This is directly linked to our ideas of conjecture inference in §7 where we hypothesise some surface as sound, but need to be able to verify this by only checking a finite sample of points on the surface. The quintessential example of this is interpolation of complex polynomials, i.e. once you know the values $P(X)$ at more than $\deg P$ points, you can determine the coefficients of P. We extend this notion to our 'higher' structure by investigating the question 'given D_1 and D_2 in $\mathrm{ZH}^!_{\mathbb{C}[X]}$, and knowing that $as(ev(D_1)) = as(ev(D_2))$ for some number of values of X and !, does $D_1 = D_2$?' The answer with one !-box and one phase variable is *yes* (Proposition 8.53), and, crucially, we can work how how many values is *enough* just from the syntax of the diagram. The phase variable part of the result (§8.2) always extends to situations with multiple phase variables, however the !-box part of the result (§8.4) cannot always be extended to multiple !-boxes, leading us to the notion of *separability* (§8.4.1). When generating hypotheses this allows us to verify hypotheses containing !-boxes where previously we could not (provided the !-boxes are separable).

For equations with phase variables but without !-boxes over the complex numbers we can actually go a step further. Theorem 8.33 allows us to verify all the phase variables in a RING$_\mathbb{C}$, ZW$_\mathbb{C}$, or ZH$_\mathbb{C}$ equation (without !-boxes) using a *single* equation without

$\mathrm{ZH}_{\mathbb{C}}$	$\mathrm{ZH}_{\mathbb{C}}$ diagrams, labelled with complex numbers
$\mathrm{ZH}_{\mathbb{C}[X]}$	$\mathrm{ZH}_{\mathbb{C}}$ with a phase variable, X
$\mathrm{ZH}_{\mathbb{C}}^{!}$	$\mathrm{ZH}_{\mathbb{C}}$ with a !-box, !
$\mathrm{ZH}_{\mathbb{C}[X]}^{!}$	$\mathrm{ZH}_{\mathbb{C}}$ with X and !
$\mathrm{Mat}_{\mathbb{C}}$	Complex matrices
$\mathbb{N} \to \mathrm{Mat}_{\mathbb{C}}$	Sequences of complex matrices
$ev : \mathrm{ZH}_{\mathbb{C}[X]} \to \mathrm{ZH}_{\mathbb{C}}$	Evaluation of X
$as : \mathrm{ZH}_{\mathbb{C}}^{!} \to \mathrm{ZH}_{\mathbb{C}}$	Assignment of a number to !
$[\![\, \cdot \,]\!] : \mathrm{ZH}_{\mathbb{C}} \to \mathrm{Mat}_{\mathbb{C}}$	Interpretation as a complex matrix

Figure 8.1: Example sets of diagrams, matrices and sequences as well as interpretations and some maps between them

any phase variables. (The ZX analogue is Theorem 8.35.) The converse to this is that whenever a phase appears in an equation less frequently than its algebraic degree it can be replaced with a variable (Remark 8.36).

This work complements the work of [Qui15], which introduced !-induction. That work allows the verification of an infinite family of equations parameterised by !-boxes, by showing an inductive step involving those !-boxes. Our work instead requires more equations to be checked, but where none of the equations involve !-boxes.

$$
\begin{array}{ccc}
\mathrm{ZH}^!_{\mathbb{C}[X]} & \xrightarrow{\;\;as\;\;} & \mathrm{ZH}_{\mathbb{C}[X]} \\
& [\![\,\cdot\,]\!] & \\
ev\big\downarrow & (\mathbb{N}\to\mathrm{Mat}_{\mathbb{C}[X]}) \longrightarrow & \mathrm{Mat}_{\mathbb{C}[X]} \quad[\![\,\cdot\,]\!] \\
\mathrm{ZH}^!_{\mathbb{C}} & \longrightarrow \mathrm{ZH}_{\mathbb{C}} & \\
[\![\,\cdot\,]\!] & (\mathbb{N}\to\mathrm{Mat}_{\mathbb{C}}) \xrightarrow{\;as\;} & \mathrm{Mat}_{\mathbb{C}}
\end{array}
$$
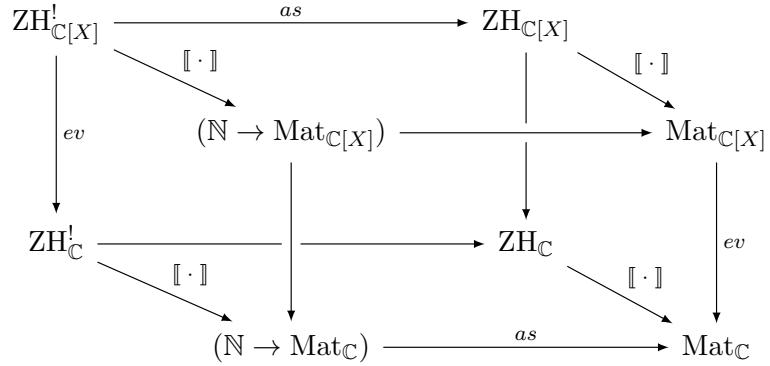
Figure 8.2: The motivational commutative cube for phase variable and !-box conjecture verification.

8.2   Verifying phase variables

Our first result will concern diagrams that contain a finite number of phase variables and no !-boxes. It relies on a certain property of polynomials: If you know the value of the polynomial $P(Y)$ for sufficiently many values of Y then you can determine all the coefficients of P. For example the polynomial $P(Y_1, Y_2) = a + bY_1 + cY_2 + dY_1Y_2$ can have all of its coefficients determined by knowing the values $P(0,0)$, $P(0,1)$, $P(1,0)$, and $P(1,1)$. We use this fact by extending the matrix interpretation of simple diagrams to one for decorated diagrams. Continuing our ZH example from §8.1 we are investigating this face of the commutative cube:

$$
\begin{array}{ccc}
\mathrm{ZH}_{\mathbb{C}[X]} & & \\
& \searrow^{[\![\,\cdot\,]\!]} & \\
ev\big\downarrow & \mathrm{Mat}_{\mathbb{C}[X]} & \\
& & \big\downarrow ev \\
\mathrm{ZH}_{\mathbb{C}} & & \\
& \searrow_{[\![\,\cdot\,]\!]} & \\
& \mathrm{Mat}_{\mathbb{C}} &
\end{array}
\tag{8.1}
$$
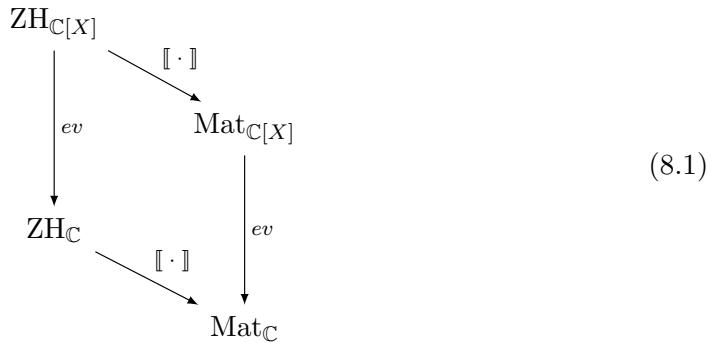
8.2.1   Matrix interpretations

The graphical calculi we have considered in this thesis all have matrix interpretations[1]. Moreover for each of these calculi a diagram with n inputs and m outputs will be mapped

---

[1] The exception being SZX in Example 2.10, and even then only to make a point.

to a matrix with $\dim \mathbb{H}^{\otimes n}$ columns and $\dim \mathbb{H}^{\otimes m}$ rows, where $\mathbb{H}$ represents R^2 (e.g. $\mathbb{C}^2$ in the case of quantum computing). We will use the language of fields and vector spaces in this chapter but note that many of the results only require the base ring to be an integral domain. A family of diagrams parameterised over α is a set of simple diagrams, and we could extend our interpretation so that a set of simple diagrams is sent to a set of matrices. This would, however, lose any structure from the phase algebra. Instead we try to find a polynomial matrix interpretation; for example one that sends a family of $ZH_\mathbb{C}$ equations $\{\mathbb{E}\}_\alpha$ to a matrix in $\mathrm{Mat}_{\mathbb{C}[\alpha]}$.

While this works well for ZH_R, ZW_R and RING_R, there is a complication with ZX: A phase variable α in a ZX diagram corresponds to an $e^{i\alpha}$ in the matrix interpretation. We can try performing the substitution $Y := e^{i\alpha}$, but run into trouble if there is a node containing, for example, $-\alpha$ (and accordingly Y^{-1} in the matrix), since polynomials do not normally allow negative powers. Rather than stick with standard polynomials we instead move to Laurent polynomials; polynomials that do allow positive and negative powers, and define all the properties we will need of them below. We do not consider the case of ZQ in this chapter, because its phase group is not commutative and commutativity is necessary for our results.

ZX also introduces one more subtlety: There is an extra relation from the phase group ($2\pi = 0$) that we should take care to reflect in our matrix interpretation. This does not impact Universal ZX, but does affect the fragments of ZX with a finite phase group (Example 8.18 demonstrates this for the Clifford+T fragment). Before we get started we define Laurent polynomials and show how to interpret diagrams with phases linear in phase variables into matrices over Laurent polynomials.

Definition 8.1 (Laurent polynomials [Art13, p356])**.** A Laurent polynomial in the variables $Y_1, \ldots, Y_n$ with coefficients in R is an element of

$$R[Y_1, Y_1', \ldots, Y_n, Y_n']/(Y_1 Y_1' = 1, \ldots, Y_n Y_n' = 1) \tag{8.2}$$

We simply write Y_j^{-1} instead of Y_j'. Laurent polynomial degrees are given by

- The 0 polynomial has degree $-\infty$ by convention

- the non-zero Laurent polynomial $a_n Y^n + a_{n-1} Y^{n-1} + \cdots + a_0 + a_{-1} Y^{-1} + \cdots + a_{-m} Y^{-m}$ with $a_n \neq 0$ and $a_{-m} \neq 0$ has *positive degree* $n \geq 0$ and *negative degree* $m \geq 0$.

Note that we can factorise this Laurent polynomial as Y^{-m} multiplied by a (non-Laurent) polynomial.

Definition 8.2 (Complex Laurent Polynomial Interpretation)**.** A *Laurent polynomial interpretation* for a graphical calculus $\mathbb{L}$ is a matrix interpretation:

$$[\![\,\cdot\,]\!] : \mathbb{L}[\alpha_1, \ldots, \alpha_n] \to \mathrm{Mat}_{R[Y_1, Y_1^{-1}, \ldots, Y_n, Y_n^{-1}]} \tag{8.3}$$

The source category here is the PROP of diagrams where generator labels are now taken not from the phase algebra of $\mathbb{L}$ but from this phase algebra adjoin the formal variables $\alpha_1, \ldots, \alpha_n$.

We give the Laurent polynomial interpretations of Universal ZX, ZW_R, ZH_R and RING_R here, showing how to link a phase variable α with the polynomial indeterminate Y. These definitions are, frankly, so close to the original that the difference is easy to miss so we show the ZX example in more detail in Example 8.3 and then the others in Example 8.4.

Example 8.3 (Laurent polynomial interpretations of Universal ZX). The Z spider from Universal ZX is parameterised by an $\alpha \in [0, 2\pi)$, and the (simple) matrix interpretation of some Z spider (with α instantiated at a) is:

$$
\left[\!\!\left[\left\{ \underset{\ldots}{\overset{\ldots}{\underset{\alpha}{\bigotimes}}} \right\}_{\alpha | \alpha = a} \right]\!\!\right] = \begin{bmatrix} 1 & 0 & \ldots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \ldots & & 0 & e^{ia} \end{bmatrix} \in \mathrm{Mat}_{\mathbb{C}} \tag{8.4}
$$

Rather than instantiate the value of α before we apply the map, we instead make the substitution $Y := e^{i\alpha}$ to find a Laurent polynomial matrix interpretation:

$$
\left[\!\!\left[\underset{\ldots}{\overset{\ldots}{\underset{\alpha}{\bigotimes}}} \right]\!\!\right] = \begin{bmatrix} 1 & 0 & \ldots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \ldots & & 0 & Y \end{bmatrix} \in \mathrm{Mat}_{\mathbb{C}[Y, Y^{-1}]} \tag{8.5}
$$

Example 8.4 (Laurent polynomial interpretations of RING, ZW and ZH). For a Laurent polynomial interpretation we simply set $\alpha = Y$ to get:

$$
\mathrm{RING}_R \qquad \overset{\ldots}{\underset{\ldots}{\underset{\alpha}{\bigotimes}}} \overset{[\![\cdot]\!]}{\longmapsto} |0 \ldots 0\rangle\langle 0 \ldots 0| + Y|1 \ldots 1\rangle\langle 1 \ldots 1| \tag{8.6}
$$

$$
\mathrm{ZW}_R \qquad \overset{\ldots}{\underset{\ldots}{\underset{\alpha}{\bigotimes}}} \overset{[\![\cdot]\!]}{\longmapsto} |0 \ldots 0\rangle\langle 0 \ldots 0| + Y|1 \ldots 1\rangle\langle 1 \ldots 1| \tag{8.7}
$$

$$
\mathrm{ZH}_R \qquad \overset{\ldots}{\underset{\ldots}{\underset{\alpha}{\boxed{}}}} \overset{[\![\cdot]\!]}{\longmapsto} \sum_{\mathrm{bitstrings}} (Y)^{i_1 \ldots i_m j_1 \ldots j_n} |j_1 \ldots j_n\rangle\langle i_1 \ldots i_m| \tag{8.8}
$$

$$(8.9)$$

The rest of the interpretations remain as usual (See Definitions 2.17, 4.38, and 4.1)

Remark 8.5. This is an unusually pedantic treatment of the interplay between interpretation and phase variables, but it illustrates the commutativity of interpretation and evaluation.

Remark 8.6. The generators for $\mathrm{ZH}_\mathbb{C}$, $\mathrm{ZW}_\mathbb{C}$ and $\mathrm{RING}_\mathbb{C}$ have Laurent polynomial matrix interpretations that are, in fact, just polynomial matrix interpretations. Really it is only for ZX that we are considering the broader Laurent polynomial version.

8.2.2   Degree of a matrix

Our hope is to find properties of the interpretations of diagrams (which are matrices of polynomials) just from the diagrams themselves. An obvious property to investigate is the degree of the polynomials in the matrix interpretation, and while we cannot easily determine this value precisely, we can find bounds for the degrees just by looking at the diagrams themselves.

Definition 8.7 (Matrix and diagram degree)**.** We define the degree of a matrix and diagram by:

- The Y_j^+-degree of a matrix in $\mathrm{Mat}_{\mathbb{C}[Y_1, Y_1^{-1}, \ldots, Y_n, Y_n^{-1}]}$ is the maximum of the positive Y_j-degrees of the entries in that matrix

- The Y_j^--degree is the maximum of the negative Y_j-degrees of the entries in that matrix

- The positive degree of a diagram is the positive degree of the matrix interpretation of that diagram (likewise for negative degrees).

- When clear from context we will refer to the degree of a phase variable α_j in the diagram, meaning the degree of Y_j in the interpretation.

Example 8.8 (Laurent polynomial and matrix degrees). Here are example positive and negative degrees for first a polynomial, and then a 2×2 matrix of polynomials.

$$\deg_Y^+ \left[\, Y^8 + 1 + Y^{-2} \,\right] = 8 \tag{8.10}$$
$$\deg_Y^- \left[\, Y^8 + 1 + Y^{-2} \,\right] = 2 \tag{8.11}$$

$$\deg_Y^+ \left[\begin{pmatrix} 2 & Y^{-3} \\ Y & Y^2 - 2 \end{pmatrix} \right] = \max\{0, 0, 1, 2\} = 2 \tag{8.12}$$

$$\deg_Y^- \left[\begin{pmatrix} 2 & Y^{-3} \\ Y & Y^2 - 2 \end{pmatrix} \right] = \max\{0, 3, 0, 0\} = 3 \tag{8.13}$$

Proposition 8.9. *For two diagrams $\mathbb{D}$ and $\mathbb{D}'$ we can find an upper bound for the degrees of the horizontal or vertical compositions of $\mathbb{D}$ and $\mathbb{D}'$, i.e.:*

$$deg^+\left[\,(\mathbb{D} \circ \mathbb{D}')\,\right] \le deg_Y^+\left[\,\mathbb{D}\,\right] + deg_Y^+\left[\,\mathbb{D}'\,\right] \tag{8.14}$$

$$deg^+\left[\,(\mathbb{D} \otimes \mathbb{D}')\,\right] \le deg_Y^+\left[\,\mathbb{D}\,\right] + deg_Y^+\left[\,\mathbb{D}'\,\right] \tag{8.15}$$

$$deg^-\left[\,(\mathbb{D} \circ \mathbb{D}')\,\right] \le deg_Y^-\left[\,\mathbb{D}\,\right] + deg_Y^-\left[\,\mathbb{D}'\,\right] \tag{8.16}$$

$$deg^-\left[\,(\mathbb{D} \otimes \mathbb{D}')\,\right] \le deg_Y^-\left[\,\mathbb{D}\,\right] + deg_Y^-\left[\,\mathbb{D}'\,\right] \tag{8.17}$$

Proof. We first note that for Laurent polynomials P and P' in $\mathbb{C}[Y, Y^{-1}]$, and for $\lambda \in \mathbb{C}$:

$$\deg_Y^+\left[\,(\lambda P)\,\right] \le \deg_Y^+\left[\,P\,\right] \tag{8.18}$$

$$\deg_Y^-\left[\,(\lambda P)\,\right] \le \deg_Y^-\left[\,P\,\right] \tag{8.19}$$

$$\deg_Y^+\left[\,(P \times P')\,\right] \le \deg_Y^+\left[\,P\,\right] + \deg_Y^+\left[\,P'\,\right] \tag{8.20}$$

$$\deg_Y^-\left[\,(P \times P')\,\right] \le \deg_Y^-\left[\,P\,\right] + \deg_Y^-\left[\,P'\,\right] \tag{8.21}$$

$$\deg_Y^+\left[\,\left(\sum_j P_j\right)\,\right] \le \max_j \deg_Y^+\left[\,P_j\,\right] \tag{8.22}$$

$$\deg_Y^-\left[\,\left(\sum_j P_j\right)\,\right] \le \max_j \deg_Y^-\left[\,P_j\,\right] \tag{8.23}$$

The composition $A \circ B$ or tensor product $A \otimes B$ of matrices produces a new matrix with entries that are linear combinations of products of the entries of A and B. Therefore:

$$\deg_Y^+\left[\,(M \circ M')\,\right] \le \deg_Y^+\left[\,M\,\right] + \deg_Y^+\left[\,M'\,\right] \tag{8.24}$$

$$\deg_Y^+\left[\,(M \otimes M')\,\right] \le \deg_Y^+\left[\,M\,\right] + \deg_Y^+\left[\,M'\,\right] \tag{8.25}$$

$$\deg_Y^-\left[\,(M \circ M')\,\right] \le \deg_Y^-\left[\,M\,\right] + \deg_Y^-\left[\,M'\,\right] \tag{8.26}$$

$$\deg_Y^-\left[\,(M \otimes M')\,\right] \le \deg_Y^-\left[\,M\,\right] + \deg_Y^-\left[\,M'\,\right] \tag{8.27}$$

Recalling that the degree of a diagram is the degree of its polynomial matrix interpretation this gives us the result for $\mathbb{D}$ and $\mathbb{D}'$. $\square$

Now that we have shown how to bound the degree of a diagram by looking at the degrees of its subdiagrams, all that is left is to calculate the degrees of the generators of the diagrams. From there we can then find bounds for the degrees of any of the diagrams

we consider in this chapter. For simplicity we will consider nodes parameterised by a single variable α, and express their degree with respect to a variable Y. Degrees of wire components are all 0.

ZX: Using $Y^n := e^{ni\alpha}$, the degrees in Y of the generators (for $n \geq 0$) are:

$$\deg_\alpha^+ \left[\boxed{n\alpha} \right] = n \qquad \deg_\alpha^- \left[\boxed{n\alpha} \right] = 0 \qquad (8.28)$$

$$\deg_\alpha^+ \left[\boxed{-n\alpha} \right] = 0 \qquad \deg_\alpha^- \left[\boxed{-n\alpha} \right] = n \qquad (8.29)$$

$$\deg_\alpha^+ \left[\boxed{n\alpha} \right] = n \qquad \deg_\alpha^- \left[\boxed{n\alpha} \right] = 0 \qquad (8.30)$$

$$\deg_\alpha^+ \left[\boxed{-n\alpha} \right] = 0 \qquad \deg_\alpha^- \left[\boxed{-n\alpha} \right] = n \qquad (8.31)$$

$$\deg_\alpha^+ \left[\; \boxed{} \; \right] = 0 \qquad \deg_\alpha^- \left[\; \boxed{} \; \right] = 0 \qquad (8.32)$$

ZH: We equate $Y := \alpha$, and for P any Laurent polynomial:

$$\deg_\alpha^+ \left[\boxed{P(\alpha)} \right] = \deg_Y^+ [\, P \,] \qquad \deg_\alpha^- \left[\boxed{P(\alpha)} \right] = \deg_Y^- [\, P \,] \qquad (8.33)$$

$$\deg_\alpha^+ \left[\bigcirc \right] = 0 \qquad \deg_\alpha^- \left[\bigcirc \right] = 0 \qquad (8.34)$$

ZW: We equate $Y := \alpha$, and for P any Laurent polynomial:

$$\deg_\alpha^+ \left[\bullet \right] = 0 \qquad \deg_\alpha^- \left[\bullet \right] = 0 \qquad (8.35)$$

$$\deg_\alpha^+ \left[\otimes \right] = 0 \qquad \deg_\alpha^- \left[\otimes \right] = 0 \qquad (8.36)$$

$$\deg_\alpha^+ \left[\ \begin{matrix} \vcenter{\hbox{$\cdots$}} \\ \boxed{P(\alpha)} \\ \vcenter{\hbox{$\cdots$}} \end{matrix}\ \right] = \deg_Y^+ [\, P\,] \qquad \deg_\alpha^- \left[\ \begin{matrix} \vcenter{\hbox{$\cdots$}} \\ \boxed{P(\alpha)} \\ \vcenter{\hbox{$\cdots$}} \end{matrix}\ \right] = \deg_Y^- [\, P\,] \tag{8.37}$$

$$\tag{8.38}$$

RING: We equate $Y := \alpha$, and for P any Laurent polynomial:

$$\deg_\alpha^+ \left[\ \begin{matrix} | \\ \boxed{P} \end{matrix}\ \right] = \deg_Y^+ [\, P\,] \qquad \deg_\alpha^- \left[\ \begin{matrix} | \\ \boxed{P} \end{matrix}\ \right] = \deg_Y^- [\, P\,] \tag{8.39}$$

$$\deg_\alpha^+ \left[\ \begin{matrix} | \\ \end{matrix}\ \right] = 0 \qquad \deg_\alpha^- \left[\ \begin{matrix} | \\ \end{matrix}\ \right] = 0 \tag{8.40}$$

$$\deg_\alpha^+ \left[\ \begin{matrix} | \\ \end{matrix}\ \right] = 0 \qquad \deg_\alpha^- \left[\ \begin{matrix} | \\ \end{matrix}\ \right] = 0 \tag{8.41}$$

$$\tag{8.42}$$

Remark 8.10. Using the interpretations above one could evaluate diagrams containing phase variables directly into polynomial rings and use this to verify an equation. This would sidestep much of the need for a verification result like the one in Theorem 8.11, but would also not reveal results like the bound on algebraic complexity in Remark 8.36 nor indicate where one could replace phases with phase variables for conjecture inference. If the direct evaluation could deal with quotient rings then it may be able to avoid pitfalls like those in Example 8.18.

8.2.3 Finite verification of phase variables

Phase variables are, as it turns out, remarkably well behaved when it comes to verification. As we will see in Theorem 8.11 we simply need to know a bound for the degree of the polynomial entries in the matrix the diagram represents. For all of the graphical calculi we have been considering this is tantamount to counting occurrences of the variable in the diagram, taking powers into account. The only other ingredient for the theorem is that of interpolation on a grid of points, a technique that is sadly not well documented as to its origin, but has existed for well over a century [Mar00].

Theorem 8.11 (Finite verification of phase variables). *For a diagrammatic equation with phase variables*

$$\{\quad \mathbb{D}_1 = \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n}$$

that has a Laurent polynomial matrix interpretation, and the equation is sound at all values of $(\alpha_1,\ldots,\alpha_n) = (a_1,\ldots,a_n) \in A_1 \times \cdots \times A_n$, where each $|A_j|$ is sufficiently large, then the equation is sound for all values of $(\alpha_1,\ldots,\alpha_n)$. That is:

$$\llbracket \{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\dots,\alpha_n | \alpha_1 = a_1,\dots,\alpha_n = a_n} \rrbracket = \llbracket \{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\dots,\alpha_n | \alpha_1 = a_1,\dots,\alpha_n = a_n} \rrbracket$$

$$\forall a_1 \in A_1, \dots, a_n \in A_n \tag{8.43}$$

$$\implies \llbracket \{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\dots,\alpha_n | \alpha_1 = a_1,\dots,\alpha_n = a_n} \rrbracket = \llbracket \{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\dots,\alpha_n | \alpha_1 = a_1,\dots,\alpha_n = a_n} \rrbracket$$

$$\forall a_1, \dots, a_n$$

The necessary size of $|A_j|$ is given by:

$$|A_j| = \max(deg^+_{Y_j}[\,\mathbb{D}_1\,], deg^+_{Y_j}[\,\mathbb{D}_2\,]) + \max(deg^-_{Y_j}[\,\mathbb{D}_1\,], deg^-_{Y_j}[\,\mathbb{D}_2\,]) + 1 \tag{8.44}$$

'The maximum of the positive degrees, plus the maximum of the negative degrees, plus one.'

Before we embark on the proof we sketch it as follows:

- Manipulate the equation $\llbracket \mathbb{D} \rrbracket = \llbracket \mathbb{D}' \rrbracket$ into an equation of the form $M = 0$, where M is a matrix of (non-Laurent) polynomials.

- Perform multivariate polynomial interpolation element-wise on M.

- In doing this interpolation we need to know an upper bound for the degrees of the polynomials in M, which we calculate using Proposition 8.9

Proof. We are seeking the multivariate complex polynomials that populate the matrices $\llbracket \mathbb{D}_1 \rrbracket$ and $\llbracket \mathbb{D}_2 \rrbracket$. We begin by combining the two matrices of Laurent polynomials into one matrix of (not-Laurent) polynomials and a scale factor of the form $Y_1^{m_1} \dots Y_n^{m_n}$.

- We define:

$$M_1 := \llbracket \mathbb{D}_1 \rrbracket \qquad M_2 := \llbracket \mathbb{D}_2 \rrbracket \tag{8.45}$$

and wish to show $M_1 = M_2$.

- First we pull enough copies of $Y_1^{-1}, \dots, Y_n^{-1}$ out of each side so that we have an equation of the form:

$$M_1' \prod_j (Y_j^{-1})^{\deg^-_{Y_j}[\,M_1\,]} = M_2' \prod_j (Y_j^{-1})^{\deg^-_{Y_j}[\,M_2\,]} \tag{8.46}$$

Where M_1' and M_2' are matrices of (not-Laurent) polynomials.

- Let $m_j := \max(\deg^-_{Y_j}[\,M_1\,], \deg^-_{Y_j}[\,M_2\,])$ and multiply both sides by $\prod_j Y_j^{m_j}$ to

clear any negative powers of Y_j.

$$M_1' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_1 \,]} = M_2' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_2 \,]} \tag{8.47}$$

- Then subtract the right hand side from the left:

$$M_1' \prod_j Y_j^{m_j - \deg_{Y_j}^- [\, M_1 \,]} - M_2' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_2 \,]} = 0 \tag{8.48}$$

$$\mathbb{M} := M_1' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_1 \,]} - M_2' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_2 \,]} \tag{8.49}$$

Note that $\mathbb{M}$ is a matrix of (not-Laurent) polynomials. The statement $\mathbb{M} = 0$ can be viewed as a stating that each entry of $\mathbb{M}$ is equal to the 0 polynomial.

- We will use the notation $\deg_{Y_j} [\, \cdot \,]$ for the degree of a (not-Laurent) polynomial, or matrix of polynomials. We could continue to use the term positive degree, the definitions coincide, but want to make it clear whenever we are not in the Laurent polynomial setting.

- We wish to find a bound for the maximum degree of any polynomial in $\mathbb{M}$:

$$\deg_{Y_j} [\, \mathbb{M} \,] \tag{8.50}$$

$$= \deg_{Y_j} \left[M_1' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_1 \,]} - M_2' \prod_j Y^{m_j - \deg_{Y_j}^- [\, M_2 \,]} \right] \tag{8.51}$$

$$\leq \max(\deg_{Y_j} [\, M_1' \,] + m_j - \deg_{Y_j}^- [\, M_1 \,], \tag{8.52}$$
$$\deg_{Y_j} [\, M_2' \,] + m_j - \deg_{Y_j}^- [\, M_2 \,])$$

$$= \max(\deg_{Y_j}^+ [\, M_1 \,] + \deg_{Y_j}^- [\, M_1 \,] + m_j - \deg_{Y_j}^- [\, M_1 \,], \tag{8.53}$$
$$\deg_{Y_j}^+ [\, M_2 \,] + \deg_{Y_j}^- [\, M_2 \,] + m_j - \deg_{Y_j}^- [\, M_2 \,])$$

$$= \max(\deg_{Y_j}^+ [\, M_1 \,] + m_j, \deg_{Y_j}^+ [\, M_2 \,] + m_j) \tag{8.54}$$

$$= \max(\deg_{Y_j}^+ [\, M_1 \,], \deg_{Y_j}^+ [\, M_2 \,]) + m_j \tag{8.55}$$

$$= \max(\deg_{Y_j}^+ [\, M_1 \,], \deg_{Y_j}^+ [\, M_2 \,]) + \max(\deg_{Y_j}^- [\, M_1 \,], \deg_{Y_j}^- [\, M_2 \,]) \tag{8.56}$$

- Suppose we know that our diagram equation is sound for parameter choices in a large enough *regular grid* of values. That is to say we know:

$$\left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n | \alpha_1 = a_1, \ldots, \alpha_n = a_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n | \alpha_1 = a_1, \ldots, \alpha_n = a_n} \right]\!\right]$$
$$\tag{8.57}$$

for $(a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$

where $|A_j| = \deg_{Y_j}(\mathbb{M}) + 1$

$A_j := \{a_{j,0}, \ldots, a_{j,\deg Y_j}\}$

By picking a polynomial entry P of $\mathbb{M}$, expressing P using the multi-index β as $P = \sum_\beta c_\beta Y^\beta$, and then evaluating P at every point in $A_1 \times \cdots \times A_n$ we construct the system of equations:

$$
\begin{bmatrix}
c_{0,\ldots,0} a_{0,\ldots,0}^{0,\ldots,0} & c_{1,\ldots,0} a_{0,\ldots,0}^{1,\ldots,0} & \cdots & c_{\deg Y_1,\ldots,\deg Y_n} a_{0,\ldots,0}^{\deg Y_1,\ldots,\deg Y_n} \\
\vdots & \vdots & \ddots & \vdots \\
c_{0,\ldots,0} a_{|A_1|,\ldots,|A_n|}^{0,\ldots,0} & c_{1,\ldots,0} a_{|A_1|,\ldots,|A_n|}^{1,\ldots,0} & \cdots & c_{\deg Y_1,\ldots,\deg Y_n} a_{|A_1|,\ldots,|A_n|}^{\deg Y_1,\ldots,\deg Y_n}
\end{bmatrix}
=
\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
\tag{8.58}
$$

Which we view as:

$$
\begin{bmatrix} V \end{bmatrix}
\begin{bmatrix} c_{0,\ldots,0} \\ \vdots \\ c_{\deg Y_1,\ldots,\deg Y_n} \end{bmatrix}
=
\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}
\tag{8.59}
$$

Where V contains all the products $a_1^{\beta_1} \times \cdots \times a_n^{\beta_n}$, β ranging from $(0, \ldots, 0)$ to $(\deg Y_1, \ldots, \deg Y_n)$. Thankfully V decomposes as:

$$
V = V_1 \otimes \ldots \otimes V_n
\tag{8.60}
$$

$$
V_j :=
\begin{bmatrix}
a_{j,0}^0 & a_{j,0}^1 & \cdots & a_{j,0}^{\deg Y_j} \\
a_{j,1}^0 & a_{j,1}^1 & \cdots & a_{j,1}^{\deg Y_j} \\
\vdots & \vdots & & \vdots \\
a_{j,\deg Y_j}^0 & a_{j,\deg Y_j}^1 & \cdots & a_{j,\deg Y_j}^{\deg Y_j}
\end{bmatrix}
\tag{8.61}
$$

- Since $\det(A \otimes B) \neq 0$ if and only if $\det(A) \neq 0$ and $\det(B) \neq 0$, and since $\det(V_j) \neq 0$ because each V_j is a Vandermonde matrix, we know that $\det(V) \neq 0$. Since V is therefore invertible we know that all the coefficients c_β must be 0, and therefore P is the 0 polynomial.

- In the presence of a regular grid on which $\mathbb{D}_1$ and $\mathbb{D}_2$ agree we know:

$$
\llbracket \{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n | \alpha_k \in A_k \forall k} \rrbracket = \llbracket \{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n | \alpha_k \in A_k \forall k} \rrbracket
\tag{8.62}
$$

$$\implies \qquad \text{for any entry } P \text{ of } \mathbb{M} \qquad P = 0 \tag{8.63}$$

$$\implies \qquad\qquad\qquad \mathbb{M} = 0 \tag{8.64}$$

$$\implies \qquad M_1' \prod_j Y^{m_j - \deg_{Y_j}^-[\,M_1\,]} = M_2' \prod_j Y^{m_j - \deg_{Y_j}^-[\,M_2\,]} \tag{8.65}$$

$$\implies \qquad\qquad\qquad M_1 = M_2 \tag{8.66}$$

$$\implies \qquad \left[\!\left[\; \{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\dots,\alpha_n} \;\right]\!\right] = \left[\!\left[\; \{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\dots,\alpha_n} \;\right]\!\right] \tag{8.67}$$

- By setting d_j to be strictly greater than $\deg_{Y_j}[\,\mathbb{M}\,]$ (which we can find a bound for using (8.56)), and $|A_j| = d_j$ we attain our result; that if we know that the interpretations of the families of diagrams agree on the regular grid described by the sizes d_j then the interpretations agree on all points in the phase algebra.

$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Box$$

Example 8.12 (Finding the sizes of the A_j). The following Universal ZX diagram contains no !-boxes and two phase variables.

$$
\begin{aligned}
&\deg_{\alpha_1}^+ \mathbb{D}_1 = 1 \quad \deg_{\alpha_2}^+ \mathbb{D}_1 = 1 \qquad \deg_{\alpha_1}^+ \mathbb{D}_2 = 1 \quad \deg_{\alpha_2}^+ \mathbb{D}_2 = 1 \\
&\deg_{\alpha_1}^- \mathbb{D}_1 = 0 \quad \deg_{\alpha_2}^- \mathbb{D}_1 = 0 \qquad \deg_{\alpha_1}^- \mathbb{D}_2 = 0 \quad \deg_{\alpha_2}^- \mathbb{D}_2 = 0
\end{aligned}
\tag{8.68}
$$

In order to apply Theorem 8.11 we should therefore construct A_1 and A_2 such that $|A_1| = \max\{1,1\} + \max\{0,0\} + 1$ and $|A_2| = \max\{1,1\} + \max\{0,0\} + 1$. By picking $A_1 = A_2 = \{0, \pi\}$ we therefore know that we can verify this parameterised family of diagram equations for all values of α_1 and α_2 by verifying this equation on the following grid of values:

$$
\begin{array}{c|cc}
 & \alpha_1 = 0 & \alpha_1 = \pi \\
\hline
\alpha_2 = 0 & (0,0) & (0,\pi) \\
\alpha_2 = \pi & (\pi,0) & (\pi,\pi)
\end{array}
\tag{8.69}
$$

I.e. by verifying the four equations:

$$\tag{8.70}$$

$$\tag{8.71}$$

we can assert that the diagram equation in (8.68) is sound for all values of α_1 and α_2 in $[0, 2\pi)$.

Remark 8.13. Example 8.12 has verified a rule that applies to all phases in $[0, 2\pi)$, but the verification used only Clifford phases. Quantum circuits using just Clifford phases are efficiently simulable on classical computers, but at time of writing Universal ZX diagrams are not [Got98].

Corollary 8.14 ([JPV18b, Theorem 3]). *In the Universal ZX calculus it suffices to check* $(\alpha_1, \ldots, \alpha_n) \in A_1 \times \cdots \times A_n$ *to prove an equation parameterised by* α_j, *where the* A_j *are sets of distinct angles with*

$$|A_j| = \max \left\{ deg_{\alpha_j}^+ [\, \mathbb{D}_1 \,], deg_{\alpha_j}^+ [\, \mathbb{D}_2 \,] \right\} + \max \left\{ deg_{\alpha_j}^- [\, \mathbb{D}_1 \,], deg_{\alpha_j}^- [\, \mathbb{D}_2 \,] \right\} + 1 \quad (8.72)$$

Remark 8.15. Corollary 8.14 was first proved in [JPV18b]. The authors of that paper use the symbol μ to count appearances of α_j (with coefficient), and T_j to denote a large enough set of values. Their method does not use Laurent polynomials, instead examining ranks of certain matrices, but this also means their method does not extend neatly to other graphical calculi.

The result of [JPV18b] is in fact stronger than that of Theorem 8.11; as they show that under the conditions given here there is a *universal* proof of the parameterised equation, something that the method given in Theorem 8.11 does not show. For the purposes of verification, however, it is not important how an equation is derived, only whether the equation is sound.

Remark 8.16 (Distances between points). It is worth noting that there is no restriction on the points in each A_j used in Theorem 8.11, beyond being distinct. If one were to work over arbitrary rings rather than fields we would require that the differences between any pair of points were not zero-divisors.

Example 8.17 (ZH phase variable verification). Note that the ZX result required the variables to be linear, but for ZH and ZW this result applies to diagrams whose phases are polynomial (or even Laurent polynomial) in the α_j. For example we can verify the following ZH equation by checking 3 distinct values of α:
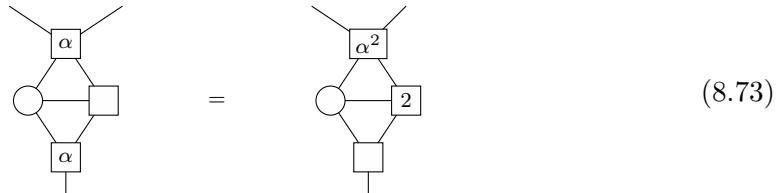
$$(8.73)$$

Example 8.18 (Phase variables in a quotient ring). Consider a Clifford+T ZX diagram that contains at least 8 nodes labelled by a positive α. Our theorem says that for any equation containing this diagram it suffices to try at least 9 distinct values of α, but this is impossible since there are only 8 distinct values of α available in Clifford+T.

This is because our Laurent polynomial matrix interpretation needs to be viewed not in $\mathbb{C}[Y, Y^{-1}]$ but in $\mathbb{C}[Y]/(Y^8 - 1)$, reflecting the property $8 \times \alpha = 0$ in our phase group. All polynomials in $\mathbb{C}[Y]/(Y^8 - 1)$ have degree at most 7, and so it is never necessary to check more than 8 points.

8.3   Phase variables over qubits

For RING$_{\mathbb{C}}$, ZW$_{\mathbb{C}}$, ZH$_{\mathbb{C}}$ and Universal ZX we can combine Theorem 8.11 with Theorem 5.42 to achieve phase variable verification with a single equation. The implication of Theorem 8.11 is that if we find 'enough' sound instantiations we can then infer that every instantiation is sound. The result of Theorem 5.42 is that we can use phase homomorphism pairs to discover more sound equations from an existing sound equation. All we therefore need to do is find the right phase homomorphisms, and the right starting equation, and we will have sufficient data to conclude soundness for a family of equations parameterised by phase variables.

In order to show this we shall require some Galois theory, which we include below, leading up to Theorem 8.33. Galois Theory concerns itself with the study of two fields, $K \subset L$, and the field automorphisms of the larger that do not affect the smaller. A classic example is the pair of fields $\mathbb{R} \subset \mathbb{C}$, where the only automorphisms of $\mathbb{C}$ that preserves $\mathbb{R}$ are the identity and complex conjugation. A second example is the pair of fields (usually just called a 'field extension') $\mathbb{Q} \subset \mathbb{Q}(\sqrt{2})$, where the only non-trivial automorphism of $\mathbb{Q}(\sqrt{2})$ that preserves $\mathbb{Q}$ is the map that sends $a + b\sqrt{2} \mapsto a - b\sqrt{2}$.

Our reason for using Galois Theory is that we will want to fix almost all of the phases in a diagram, while affecting some others. For example in a ZX diagram with phases that are multiples of $\pi/8$ we can define the fields $K := \mathbb{Q}(e^{i\pi/4})$ and $L := \mathbb{Q}(e^{i\pi/8})$. The automorphisms of L that fix K will lift to phase automorphisms that affect only those phases that are odd multiples of $\pi/8$. Much of the work below is in proving the existence of suitable field extensions, and then showing that suitable automorphisms exist. Definitions, lemmas, and theorems leading up to Theorem 8.33 are either directly from the literature (and cited) or are expected to be well know (and a reference could not be found). Theorems 8.33 and 8.35 are the crux of this section. This section concerns itself with qubit graphical calculi, and as such all the fields considered will be subfields of $\mathbb{C}$.

Definition 8.19 (Degree of a field extension [Ste15, Definition 6.2])**.** Where K is a subfield of L, written L/K, we define the degree of the extension $[L : K]$ as the dimension of L considered as a vector space over K.

Lemma 8.20 ([Ste15, Proposition 6.7])**.** *For $K(\alpha)/K$ a field extension where α satisfies some minimal polynomial $m \in K[X]$,*

$$[K(\alpha) : K] = \deg m \tag{8.74}$$

Definition 8.21 (Normal extension [Ste15, Definition 9.8])**.** The field extension L/K is called *normal* if every irreducible polynomial over K with a root in L splits in L as a product of linear factors.

Theorem 8.22 (Normal closure [Ste15, Theorem 11.6])**.** *If L/K is a finite extension of subfields of $\mathbb{C}$, then there exists a unique smallest normal extension $N/L/K$ which is a finite extension of K.*

Theorem 8.23 (Tower Law [Ste15, Corollary 6.6])**.** *If $K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n$ are subfields of $\mathbb{C}$, then*

$$[K_n : K_0] = [K_n : K_{n-1}][K_{n-1} : K_{n-2}]\ldots[K_1 : K_0] \tag{8.75}$$

Theorem 8.24 (Part of the fundamental theorem of Galois theory [Ste15, Theorem 12.2])**.** *If L/K is a finite, normal field extension inside $\mathbb{C}$ then $Gal(L/K)$, the group of field automorphisms of L that fix K, has order $[L : K]$*

We will need to be able to place an upper bound on the degree of an algebraic extension, in a similar way to how we needed to place an upper bound on the degree of a matrix. Assuming we know for each α a polynomial p where $p(\alpha) = 0$, then we know that the degree of the minimal polynomial m_α has lower degree than p. We can use this, in conjunction with the following corollary, to get an upper bound for the degree of an algebraic extension.

Corollary 8.25. *For $\beta_1, \ldots, \beta_r$ algebraic over K, and m_{β_j} the minimal polynomial of β_j over K*

$$[K(\beta_1, \ldots, \beta_r) : K] \leq \prod_j \deg m_{\beta_j} \tag{8.76}$$

Now that we know how to find certain bounds we shall also need to find extensions that are guaranteed to exceed those bounds. For this we shall use primitive roots of unity and cyclotomic polynomials to introduce into a field K elements that guarantee $[K(\alpha) : K] \geq d$ for a given d.

Definition 8.26 (Primitive Roots of Unity [Ste15, Definition 1.5])**.** For $n \in \mathbb{N}$ the primitive n-th roots of unity are

$$\left\{ e^{2ij\pi/n} | (j, n) = 1 \right\} \tag{8.77}$$

and we shall use ω_n to indicate a generic primitive n-th root of unity.

Definition 8.27 (Cyclotomic polynomials [Ste15, Definition 21.5])**.** The n-th cyclotomic polynomial Φ_n is defined as:

$$\Phi_n(X) := \prod_{1 \leq a \leq n, \ (a,n)=1} (X - \omega_n^a) \tag{8.78}$$

for ω_n a primitive n-th root of unity

Lemma 8.28 (Galois group of a cyclotomic extension [Ste15, Theorem 21.9])**.** *The Galois group of $\mathbb{Q}(\omega_n)/\mathbb{Q}$ is isomorphic to $(\mathbb{Z}/n\mathbb{Z})^*$, and the degree of Φ_n is $|(\mathbb{Z}/n\mathbb{Z})^*|$*

Lemma 8.29 (Φ_m irreducible over transcendental extension)**.** *Φ_m is irreducible over $\mathbb{Q}(\pi_1, \ldots, \pi_n)$, where $n \geq 0$ and the π_j are transcendental and algebraically independent over $\mathbb{Q}$.*

Proof. If $n = 0$ then Φ_m is irreducible ([Ste15, Corollary 21.6]). If $n > 0$ then let us assume that we can factorise Φ_m, i.e. for ω_m a primitive m-th root of unity there is a polynomial f such that

$$f(\omega_m) = 0 \qquad \deg f < \deg \Phi_m \qquad f(X) \in \mathbb{Q}(\pi_1, \ldots, \pi_n) \tag{8.79}$$

and without loss of generality we can assume that π_n is used non-trivially as a coefficient of f. We can modify f by clearing denominators and negative powers of π_n to get

$$g(\pi_n) = 0 \qquad \deg g < \infty \qquad g(X) \in \mathbb{Q}(\pi_1, \ldots, \pi_{n-1}, \omega_m) \tag{8.80}$$

We therefore can place upper bounds on the degrees of the relevant field extensions:

$$[\mathbb{Q}(\pi_1, \ldots, \pi_n, \omega_m) : \mathbb{Q}(\pi_1, \ldots, \pi_{n-1}, \omega_m)] \leq \deg g \tag{8.81}$$

$$[\mathbb{Q}(\pi_1, \ldots, \pi_{n-1}, \omega_m) : \mathbb{Q}(\pi_1, \ldots, \pi_{n-1})] \leq \deg \Phi_m \tag{8.82}$$

Therefore $[\mathbb{Q}(\pi_1, \ldots, \pi_n, \omega_m) : \mathbb{Q}(\pi_1, \ldots, \pi_{n-1})]$ is finite, and therefore $[\mathbb{Q}(\pi_1, \ldots, \pi_n) : \mathbb{Q}(\pi_1, \ldots, \pi_{n-1})]$ is finite, contradicting the algebraic independence of $\{\pi_j\}$ over $\mathbb{Q}$. $\square$

Lemma 8.30. *For K a finitely generated field extension of $\mathbb{Q}$, and any choice of natural number d, there exist infinitely many primes p such that $[K(\omega_p) : K] \geq d$*

Proof. We decompose $K/\mathbb{Q}$ as $K/L/\mathbb{Q}$ where $L/\mathbb{Q}$ is transcendental (and finitely generated by algebraically independent additions to $\mathbb{Q}$) and K/L is algebraic (and finitely generated by algebraically dependent additions to L).

By Lemma 8.29 Φ_m is irreducible in L, and therefore $[L(\omega_m) : L] = \deg \Phi_m$. Let r be $[K : L]$, and s be $[K(\omega_m) : K]$. Note that r is fixed and we are trying to choose m such that s is larger than d. We illustrate this with the diagram of extensions:

$$\tag{8.83}$$

We know that

$$[K(\omega_m) : L] = [K(\omega_m) : L(\omega_m)][L(\omega_m) : L] \geq \deg \Phi_m \qquad (8.84)$$

$$[K(\omega_m) : L] = [K(\omega_m) : K][K : L] = rs \qquad (8.85)$$

The degree of Φ_p for p a prime is $p - 1$, therefore by choosing $m = p$ for $p > rd$ we know that $[K(\omega_m) : K] \geq \deg \Phi_p / r \geq d$ $\qquad\qquad\qquad\square$

Lemma 8.31. *For the finite set of distinct primes $\{p_j\}$ there is an isomorphism of groups*

$$Gal(K(\omega_{p_1}, \ldots, \omega_{p_n})/K) \cong \prod_j Gal(K(\omega_{p_j})/K) \qquad (8.86)$$

Proof. Because the roots of Φ_p are all powers of any other root, either all the roots of Φ_p are in K or none of them are. So either ω_p is in K and $\mathrm{Gal}(K(\omega_p)/K) \cong \{e\}$ or ω_p is not in K and $\mathrm{Gal}(K(\omega_p)/K) \cong (\mathbb{Z}/p\mathbb{Z})^*$. We can therefore ignore any $\omega_p \in K$ as it contributes trivially to both $\mathrm{Gal}(K(\omega_{p_1}, \ldots, \omega_{p_n})/K)$ and $\prod_j \mathrm{Gal}(K(\omega_{p_j})/K)$.

Only considering those $\omega_p \notin K$, Lemma 8.28 gives us the isomorphism

$$\mathrm{Gal}(K(\omega_p)/K) \cong (\mathbb{Z}/p\mathbb{Z})^* \qquad (8.87)$$

$$(\sigma : \omega_p \mapsto \omega_p^k) \leftrightarrow (k \mod p) \qquad (8.88)$$

By writing $N := \prod_j p_j$ and $\Omega := \prod_j \omega_{p_j}$, and noting that $K(\omega_{p_1}, \ldots, \omega_{p_n}) = K(\Omega)$, we construct the isomorphism we need via the Chinese Remainder Theorem:

$$\mathrm{Gal}(K(\Omega)/K) \cong \prod_j \mathrm{Gal}(K(\omega_{p_j})/K) \qquad (8.89)$$

$$(\sigma : w_N \mapsto w_N^k) \leftrightarrow (\sigma_1 : \omega_{p_1} \mapsto \omega_{p_1}^{(k \mod p_1)}, \ldots, \sigma_1 : \omega_{p_n} \mapsto \omega_{p_n}^{(k \mod p_n)}) \qquad (8.90)$$

$$(k \mod N) \leftrightarrow (k \mod p_1, \ldots, k \mod p_n) \qquad (8.91)$$

$$\square$$

Lemma 8.32. *For any list of natural numbers $d_1, \ldots, d_n$ and finitely generated field extension K of $\mathbb{Q}$ we can construct elements $a_1, \ldots, a_n$ of $\mathbb{C}$ (which happen to be roots of unity) such that*

$$Gal(K(a_1, \ldots, a_n)/K) \cong Gal(K(a_1)/K) \times \cdots \times Gal(K(a_n)/K) \qquad (8.92)$$

$$|Gal(K(a_j)/K)| \geq d_j \qquad \forall j \qquad (8.93)$$

Proof. For any natural number d there exists a prime p such that $[K(\omega_p) : K] \geq d$ by

Lemma 8.30. Therefore by Lemma 8.31

$$\mathrm{Gal}(K(\omega_{p_1}, \ldots, w_{p_n})/K) \cong \mathrm{Gal}(K(\omega_{p_1})/K) \times \cdots \times \mathrm{Gal}(K(\omega_{p_n})/K) \tag{8.94}$$

and for each subgroup $|\mathrm{Gal}(K(\omega_{p_j})/K)| \geq d$. Set a_j to a choice of ω_{p_j}, such as $e^{2i\pi/p_j}$. $\square$

Theorem 8.33 (Single equation verification of qubit phase variables, ring version)**.** *For a* RING$_\mathbb{C}$*, ZW$_\mathbb{C}$, or ZH$_\mathbb{C}$ equation with phases that are polynomial in some phase variables, there exists a single verifying equation without any phase variables. If we can put an upper bound on the degrees of the minimal polynomials of the phase constants then we can construct such an equation.*

Proof. Let $D_1 = D_2$ be the equation of interest, with phase variables $\alpha_1, \ldots, \alpha_n$, and phase constants $\{c_1, \ldots, c_m\}$. Let K be the normal closure of $\mathbb{Q}(c_1, \ldots, c_m)$ in $\mathbb{C}$; this exists by Theorem 8.22, is finite over $\mathbb{Q}(c_1, \ldots, c_m)$, and is therefore finitely generated over $\mathbb{Q}$ by the $\{c_1, \ldots, c_m\}$ and finitely many more elements (any missing Galois conjugates). Note that K satisfies the conditions of Lemma 8.30. We will construct the assignments

$$\alpha_1 = a_1, \alpha_2 = a_2, \ldots, \alpha_n = a_n \tag{8.95}$$

and then show that this instantiation leads to the desired outcome.

$$
\begin{array}{c}
K(a_1, \ldots, a_n) \\
\Big| \text{large enough} \\
\mathrm{K} \\
\Big| \text{normal closure} \\
\mathbb{Q}(c_1, \ldots, c_m) \\
\Big| \\
\mathbb{Q}
\end{array}
\tag{8.96}
$$

By Lemma 8.32 there exist elements $a_1, \ldots, a_n \in \mathbb{C}$ such that

$$\mathrm{Gal}(K(a_1, \ldots, a_n)/K) \cong \mathrm{Gal}(K(a_1)/K) \times \cdots \times \mathrm{Gal}(K(a_n)/K) \tag{8.97}$$

and $|\mathrm{Gal}(K(a_j)/K)| \geq d_j$ for all j; note that d_j is the required size of the sets A_j, as in Theorem 8.11.

In order to construct the phases we shall need to know some algebraic information about all the phases c_j already present in the diagram. That is for a given c_j we will need to know either that c_j is transcendental over $\mathbb{Q}$, or some polynomial that c_j satisfies. We can then apply Corollary 8.25 to get an upper bound on the degree of the algebraic extension. Then using Lemma 8.32 we can construct suitable values $a_1, \ldots, a_n$.

Suppose the instantiation $\alpha_1 = a_1, \ldots, \alpha_n = a_n$ of $D_1 = D_2$ is sound. By Theorem 5.42 we know that applying $\hat{\phi}$ to this instantiation is also sound, for ϕ an auto-

morphism that fixes K.

Therefore for any element ϕ of $\mathrm{Gal}(K(a_1,\ldots,a_n)/K)$ we know that the instantiation $\alpha_1 = \phi(a_1),\ldots,\alpha_n = \phi(a_n)$ is sound. Since the Galois group splits as a product of groups each affecting a single a_j (Lemma 8.32) we can apply Theorem 8.11 with the sets:

$$A_j := \{\phi(a_j)|\phi \in \mathrm{Gal}(K(a_j)/K)\} \tag{8.98}$$

knowing that each of these choices of instantiations yields a sound equation. Therefore by verifying the phase-variable-parameterised equation $D_1 = D_2$ with the instantiation $\alpha_1 = a_1,\ldots,\alpha_n = a_n$ we have verified $D_1 = D_2$ for all values of the variables. $\square$

Example 8.34 (Verifying a $\mathrm{RING}_{\mathbb{C}}$ family of equations with a single equation). Consider the $\mathrm{RING}_{\mathbb{C}[x,y]}$ equation

$$\tag{8.99}$$
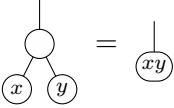
By Theorem 8.11 we require sets A_x and A_y, with sizes $d_x \geq 2$ and $d_y \geq 2$. We can therefore instantiate x to $e^{2i\pi/3}$ and y to $e^{2i\pi/5}$, knowing that

$$\text{sound} \tag{8.100}$$
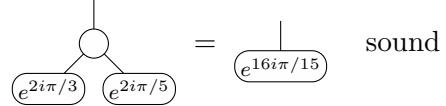
$$\Longrightarrow \qquad \text{sound } \forall\phi \in \mathrm{Gal}(\mathbb{Q}(\omega_3,\omega_5)/\mathbb{Q}) \tag{8.101}$$
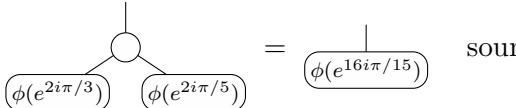
$$\Longrightarrow \qquad \text{sound } \forall x,y \in \mathbb{C} \text{ by Theorem 8.11} \tag{8.102}$$
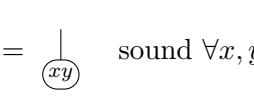
In the language of §7: The equation

$$\tag{8.103}$$
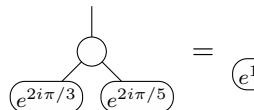
implies and is implied by the equation

$$\left\{ \quad \right\}_{x,y} \quad x,y \text{ phase variables} \tag{8.104}$$
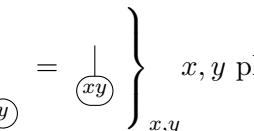
Theorem 8.35 (Single equation verification of qubit phase variables, ZX version)**.** *For*

a Universal ZX equation with phases that are linear in some phase variables, there exists a single verifying equation without any phase variables. If all the phase constants are rational multiples of π then we can construct such an equation.

Proof. The idea of the proof of Theorem 8.35 is nearly identical to that of Theorem 8.33. The difference is that we cannot rely directly on the results about phase homomorphisms and so must reconstruct the necessary properties directly.

We first regard the ZX equation as generated by Z spiders and Hadamard gates. Each concrete phase c on a green node corresponds to e^{ic} in the matrix representation. We therefore construct K as the normal closure of $\mathbb{Q}(e^{ic_1}, \ldots, e^{ic_m}, e^{i\pi/4})$. The $e^{i\pi/4}$ ensures that $\sqrt{2}$ is in K and so the automorphisms created will not unintentionally interact with the interpretation of the Hadamard gate. By Lemma 8.32 there exist roots of unity $a_1, \ldots, a_n \in \mathbb{C}$ such that

$$\mathrm{Gal}(K(a_1, \ldots, a_n)/K) \cong \mathrm{Gal}(K(a_1)/K) \times \cdots \times \mathrm{Gal}(K(a_n)/K) \qquad (8.105)$$

and $|\mathrm{Gal}(K(a_j)/K)| \geq d_j$ for all j; note that d_j is the required size of the sets A_j, as in Theorem 8.11.

Any given $\phi \in \mathrm{Gal}(K(a_1, \ldots, a_n)/K)$ sends $\{a_1, \ldots, a_n\}$ to $\{b_1, \ldots, b_n\}$ where b_j is a Galois conjugate of a_j and can be viewed as conjugating the phases that each root of unity represents (via $a \leftrightarrow e^{i\alpha}$). We lift this permutation to a map $\hat{\phi}$ for the ZX diagrams, noting that this is a very similar construction to that of phase group homomorphism pairs. Since K is fixed by ϕ we know that $\hat{\phi}$ does not change any of the concrete phases from the initial diagram. Since $e^{i\pi/4}$ is in K we can simply check that for Z spiders and for Hadamard gates that ϕ and $\hat{\phi}$ respect $[\![\cdot]\!]$ in the sense that:

$$\phi([\![\cdot]\!]) = [\![\hat{\phi}(\cdot)]\!] \qquad (8.106)$$

This gives us the property that if our original equation was sound then so is the image under $\hat{\phi}$. We can therefore again apply Theorem 8.11 to show that the constructed equation (with phases $\{q_j\}$ instead of variables) and all the images of this equation under $\hat{\phi}$ verify the original family of parameterised equations.

The argument about construction is also similar to that in the proof of Theorem 8.33, noting that the Universal ZX phase $2a\pi/b$ with a/b rational represents the complex number $e^{i2a\pi/b}$, which is a solution to the equation $X^b - 1 = 0$, and so the degree of the corresponding minimal polynomial is bounded above by b. $\qquad \square$

Remark 8.36. One does not have to work from parameterised equation to verifying, instantiated equation. The presence of a phase τ in $\mathbb{A} \setminus \mathbb{Q}$ for a RING$_\mathbb{C}$ equation already suggests at least two sound equations (since there is a non-identity Galois automorphism ϕ for τ over $\mathbb{Q}$, and we know that $\hat{\phi}$ preserves soundness). Such a phase τ suggests, in the conjecture inference sense, the family of equations where τ has been replaced by a

phase variable. Again this is similar to the replacement of existential quantifiers with universal quantifiers as mentioned in §3.2.

One could also view this result as a bound on the 'algebraic complexity' of phases in an equation of diagrams: If any field element appears less frequently than the degree of its minimal polynomial then we should seek to replace it with a phase variable. One does, however, need to be careful when trying to calculate how often an element appears in a diagram. For example $\sqrt{6}$ and $\sqrt{2}$ are not algebraically independent, so naively counting $\sqrt{2}$ as not occurring in $\sqrt{6}$ will lead to errors.

Having brought the number of verifying equations needed down to one (in the qubit case) we now turn our attention to !-boxes, which are only marginally less well behaved.

8.4   Verifying !-boxes

!-boxes allow the succinct representation of arbitrary numbers of copies of subdiagrams. In this section we will show that unless two !-boxes are connected (and not nested) then this family of diagrams will behave in a very constrained manner. The main result is that if an equation includes !-boxes that are *separable* (i.e. not connected as described above) then we need only check the first N instances of the !-box expansions in order to verify the entire family. As with the result on phase variable verification we can read this value N off the diagram directly. Part of this result is the definition of *series !-box form*, a manipulation of the diagram (requiring spider laws) that presents the !-box expansions in a novel manner. This manipulation allows us to view each further !-box expansion in terms of the ∘ product rather than the more traditional ⊗ presentation.

In terms of our running ZH example from §8.1 we are now investigating this face of the commutative cube:

$$
\begin{array}{ccc}
\mathrm{ZH}_{\mathbb{C}}^{!} & \xrightarrow{\ as\ } & \mathrm{ZH}_{\mathbb{C}} \\[2pt]
\llbracket \cdot \rrbracket \searrow & & \searrow \llbracket \cdot \rrbracket \\[2pt]
(\mathbb{N} \to \mathrm{Mat}_{\mathbb{C}}) & \xrightarrow{\ as\ } & \mathrm{Mat}_{\mathbb{C}}
\end{array}
\tag{8.107}
$$

With a third face given in (8.108).

8.4.1   Copies, nesting, and separability

There is a choice to be made when expanding a !-box that contains a phase variable. The approach taken in [Mer14], which we adopt here, is to copy the variable name exactly. This is in contrast to creating a 'fresh' name for each new instance of the variable (Quantomatic, Ref. [KZ15], implements a version of this for users that want it). The original paper that this work appeared in, [Mil20], contains versions of these proofs that encompass both options, but since only minor alterations are required at any point we now present just the proof for the following definition of copying:

Definition 8.37 (Copying variables inside !-boxes)**.** When a !-box expansion creates new instances of a phase variable we *copy* that variable name, so that all instances are linked by the same name (the approach taken in [Mer14, §4.4.2])

This definition results in the commutativity of *ev* and *as*, giving us another face in the commutative cube from our running example (see §8.1).

$$
\begin{array}{ccc}
\mathrm{ZH}^!_{\mathbb{C}[X]} & \xrightarrow{\;as\;} & \mathrm{ZH}_{\mathbb{C}[X]} \\
\downarrow{ev} & & \downarrow{ev} \\
\mathrm{ZH}^!_{\mathbb{C}} & \xrightarrow{\;as\;} & \mathrm{ZH}_{\mathbb{C}}
\end{array}
\tag{8.108}
$$

Definition 8.38 (nesting order)**.** We define a partial order, called the nesting order, on !-boxes in a diagram:

$$\delta_1 < \delta_2 \quad \text{if } \delta_1 \text{ is inside } \delta_2 \tag{8.109}$$

And use this partial order to draw a nesting diagram. For example this Universal ZX diagram:

 has nesting diagram $\begin{array}{cc} \delta_1 & \delta_2 \\ \downarrow & \\ \delta_3 & \end{array}$ (8.110)
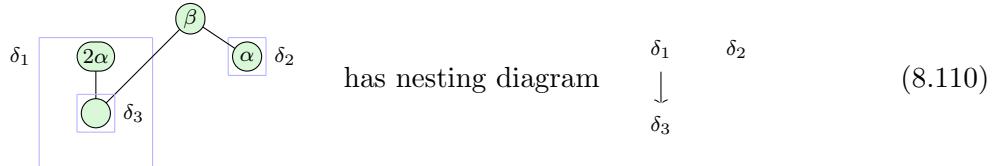
Remark 8.39. The nesting diagram is the same as just taking the !-nodes in a pattern graph.

Definition 8.40 (Well nested)**.** We say an equation is *well nested* if the nesting diagrams corresponding to the left and right hand sides of the equation are identical.

Definition 8.41 (Join)**.** The *join* of a !-box is the collection of wires that leave that !-box, i.e. the edges linking a vertex inside δ to one outside δ. The *size* of a join is the number of wires, and the *dimension* of a join is the dimension of the diagram formed by just the wires of that join, i.e. $\dim \left[\!\!\left[\; \middle| \; ^{\otimes n} \right]\!\!\right]$ where n is the size of the join. This is equivalently $\left(\dim \left[\!\!\left[\; \middle| \; \right]\!\!\right] \right)^n$.

Definition 8.42. We describe a pair of !-boxes as **separated** if either:

- They are nested, or

- There is no edge joining a vertex in one to a vertex in the other

Theorem 8.51 will require that the !-boxes all be separated, but before we look at that theorem we will investigate how stringent a requirement that is; or rather the question 'if two !-boxes are not separated, then how hard is it to separate them?' This question is directly relevant for conjecture synthesis, as it allows us to convert an equation that is not amenable to finite verification into an equation that is amenable to finite verification. This determines which sorts of conjectures we should generate in our conjecture synthesis.

Definition 8.43 (Separable)**.** We describe a non-separated pair of !-boxes as separable if we can perform the following operation:

$$(8.111)$$

We define pairs of nodes as separable if we can always separate !-boxes that are joined by edges between these pairs of nodes. Note that we only need to consider nodes that have arbitrary arity, since only they can be connected to !-boxes.

Lemma 8.44. *The following pairs of nodes are separable, by calculus:*

$$ZX: \quad (8.112)$$

$$ZH: \quad (8.113)$$

$$ZW: \quad (8.114)$$

$$\textsc{ring}: \quad (8.115)$$

Proof. These proofs follow immediately from the spider and bialgebra laws in the respective calculi. Note that it is enough to specify the phase-free versions of these interactions, because phases can always be moved away from the critical nodes. $\qquad\square$

Having shown how to separate !-boxes from each other, we will now talk about how to spread apart the instances of a !-box once it has been instantiated. The form we are aiming for is quite particular; one that allows us to use properties of the vertical composition ∘.

8.4.2   Series !-box form

Instantiation of !-boxes is an operation of pattern graphs and it is only because the calculi we are considering are built from spiders that the instantiated outcomes still form valid diagrams. After instantiating !-boxes, and in the presence of suitable spider laws, we can change the presentation of the repeated elements into more amenable forms. In particular we can represent the d copies created by a !-box as a d-fold composition of diagrammatic elements, within the context of a larger diagram. We call this 'series !-box form', providing a definition, example and method below.

Definition 8.45 (Series !-box form). *Series !-box form* for a given non-nested, separated !-box δ_1 is a presentation $C \circ (B^{\circ d}) \circ G$ of each the $(\delta_1 = d)$-instantiated diagrams, where

- $C :=$ An unparameterised end diagram

- $B :=$ Repeated element, which may contain $\alpha_j \forall j$, $\delta_k \forall k \geq 2$ and some boundary nodes

- $G :=$ The rest of the diagram outside of δ_1, which may contain $\alpha_j \forall j$, $\delta_k \forall k \geq 2$ and some boundary nodes

Example 8.46 (Series !-box form). To give a simple ZX example of series !-box form, with just one !-box and just one wire between G and B':

$$(8.116)$$

Note that B is not just the subdiagram B' but also everything directly below it as depicted in (8.116). Although we have only used one example node and joining wire we can perform this action on all nodes and joining wires. Where two wires travel from the !-box to the same spider inside G we first spread out that spider so each wire from the !-box connects to a different spider in G. Here is an example for n wires between B' and G, and p boundaries inside δ.

n end nodes d instances of B n nodes inside G

(8.117)
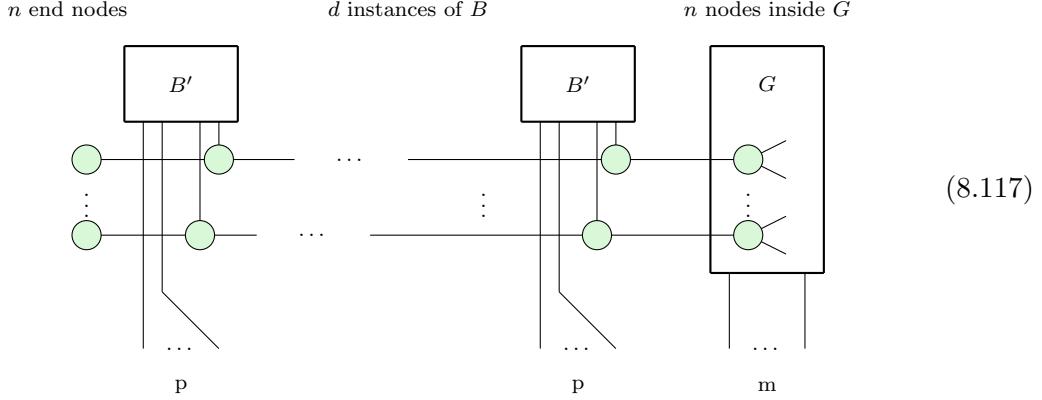
From here it is easy to see that we have the diagram $G : m \to n$, beside d copies of a diagram we call $B : p+n \to n$ (containing the p boundary nodes and the new connecting spiders), and finally an ending diagram $C : n \to 0$.

Proposition 8.47. *In the calculi ZX, ZW and ZH, for any diagram $\mathbb{D}$ and any value of d we can put $\mathbb{D}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d}$ into series !-box form.*

Proof. We will only indicate the !-box we are interested in (δ) and not explicitly write out the other parameters. First we manipulate the diagram (by Choi-Jamiolkowski we can transform inputs into outputs, provided we turn them back again later) until it is in the following form:

$$\mathbb{D} \quad = \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (8.118)$$

p boundary nodes for B

!-box δ around subdiagram B'

n connecting wires from B to G

remaining diagram G

m boundary nodes for G
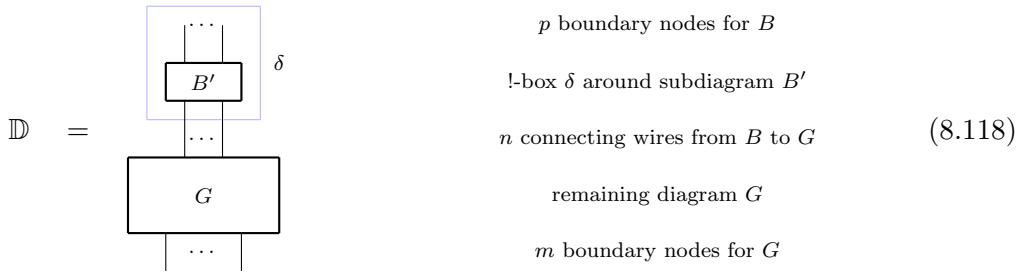
We note that the nodes inside G that join with B' must be spiders; since there can be arbitrary many instances of B' the connecting node in G must be able to have arbitrary arity. There may also be p boundaries that are contained in δ. We will now rely on the existence of a spider law such that we may do the following:

$$(8.119)$$
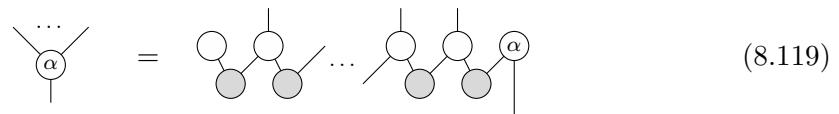
Which is the ability to 'spread' a spider with q outputs into q repeated copies of a spider

with 1 output, with suitable initial, terminal, and joining subdiagrams. Such spider laws exists for ZW, ZX, ZH and RING. We apply this to all the spiders in G that are connected to the instantiated subdiagrams formed by δ, so that the spiders form a chain as in (8.117). □

8.4.3   The verification process for !-boxes

Using this notion of series !-box form (Definition 8.45) we will now show that checking just an initial segment of a !-box instantiation suffices for showing soundness for all possible instantiations. Before stating and proving our theorem we will first need two simple lemmas concerning tensor products of bases, and increasing chains of vector subspaces.

Lemma 8.48. *The set of all vectors of the form $\{v_d \otimes \ldots \otimes v_1 \otimes x\}$ where $v_j \in \mathbb{H}^{\otimes p}$ and $x \in \mathbb{H}^{\otimes m}$ contains a basis for $\mathbb{H}^{\otimes(m+dp)}$*

Proof. Note that we may form a basis for $V \otimes V'$ by taking the tensor products of the bases of V and V', and therefore the above set contains all the basis elements of $\mathbb{H}^{\otimes p} \otimes \ldots \mathbb{H}^{\otimes p} \otimes \mathbb{H}^{\otimes m} \cong \mathbb{H}^{\otimes(m+dp)}$. □

Lemma 8.49. *A chain of subspaces V_j of an N-dimensional vector space V' has the following property:*

$$If$$
$$V_j \leq V_{j+1} \ \forall j \tag{8.120}$$
$$(V_j = V_{j+1}) \implies (V_{j+1} = V_{j+2}) \tag{8.121}$$
$$Then$$
$$\exists n \leq N \, s.t. \ V_{n+k} = V_n \ \forall k \in \mathbb{N} \tag{8.122}$$

Proof. By the assumptions in the statement V_j is non-decreasing, $\dim V_j \leq N$, and if $V_j = V_{j+1}$ then $V_j = V_k \forall k \geq j$; therefore the sequence $\{V_j\}$ has an initial increasing section, followed by a stable section. By looking at the sequence $\{\dim V_j\}$ we know that this initial increasing section has length at most N. □

Definition 8.50 (Verifying subset)**.** Given a parameterised equation $\mathbb{E}$ we say that $\{\mathbb{E}_1, \ldots, \mathbb{E}_n\}$ verifies $\mathbb{E}$ if:

$$(\forall j \ \mathbb{E}_j \text{ is sound}) \implies \mathbb{E} \text{ is sound}$$

Theorem 8.51 (Finite verification of !-boxes)**.** *Given a family*

$$\mathbb{E} = \{ \quad \mathbb{D}_1 = \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m}$$

of diagrammatic equations, in ZX, ZH, or ZW, parameterised by a !-box δ_1 where δ_1 is separated from all other !-boxes and is nested in no other !-box; then $\mathbb{E}$ is verified by the finite family

$$\{\mathbb{E}|_{\delta_1=0}, \ldots, \mathbb{E}|_{\delta_1=N}\}$$

where N is the dimension of the join of δ_1 in $\mathbb{D}_1$ plus the dimension of the join of δ_1 in $\mathbb{D}_2$. That is:

$$n_1 := join\ of\ \delta_1\ in\ \mathbb{D}_1$$
$$n_2 := join\ of\ \delta_1\ in\ \mathbb{D}_2$$
$$N := \dim\left(\mathbb{H}^{\otimes n_1} \oplus \mathbb{H}^{\otimes n_2}\right)$$

$$\left[\!\!\left[\ \{\quad \mathbb{D}_1 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d_1}\ \right]\!\!\right]$$
$$= \left[\!\!\left[\ \{\quad \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d_1}\ \right]\!\!\right] \qquad \forall d_1 \leq N \qquad (8.123)$$

$$\implies \left[\!\!\left[\ \{\quad \mathbb{D}_1 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m}\ \right]\!\!\right]$$
$$= \left[\!\!\left[\ \{\quad \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m}\ \right]\!\!\right] \qquad (8.124)$$

The sketch of the proof is:

1. Manipulate the diagrams into series !-box form (Definition 8.45)

2. Move to the matrix interpretation

3. Manipulate the equation between two matrices into an expression on a single vector space of dimension N

4. Demonstrate the required property as a condition on subspaces

Proof. Considering either $\mathbb{D}_1$ or $\mathbb{D}_2$ for now, manipulate $\mathbb{D}_{\delta_1=d}$ into series !-box form; $C \circ (B)^{\circ d} \circ G$. The p boundary vertices in every copy of B pose an issue for our intended method, and so we instead consider B as being parameterised not just by $\alpha_j\ \forall j$ and $\delta_k\ \forall k > 1$, but also by input vectors $v \in \mathbb{H}^{\otimes p}$ that 'plug' the inputs inside B.

Since we can show equivalence of complex matrices by showing that they perform the same operation on any input, it will be enough to show that for any choice of α_j, $\delta_{k>1}$ and v that (8.124) holds. Assuming we have instantiated the α_j and $\delta_{k>1}$, Lemma 8.48 justifies that we can choose the input vector for the p inputs of each of the d copies of B independently.

Given a choice of values for the α_j, $\delta_{k>1}$ and v we denote this choice by q and use B_q to mean 'the sub-diagram B from the series !-box form with this choice of variables'. We

also define G_q as the subdiagram G instantiated with the values of α_j and $\delta_{k>1}$ described by q, same as for B_q (we do not plug any values into the inputs of G). Once we have chosen values for q we may consider the matrix interpretation of the diagram:

$$[\![\, C \,]\!]\, [\![\, B_{q_d} \,]\!] \ldots [\![\, B_{q_1} \,]\!]\, [\![\, G_{q_o} \,]\!] \tag{8.125}$$

$$G_q : \mathbb{H}^{\otimes m} \to \mathbb{H}^{\otimes n} \tag{8.126}$$

$$B_q : \mathbb{H}^{\otimes n} \to \mathbb{H}^{\otimes n} \tag{8.127}$$

$$C : \mathbb{H}^{\otimes n} \to \mathbb{C} \tag{8.128}$$

Going back to our original question: Given an equation $\mathbb{D}_1 = \mathbb{D}_2$ of two families of diagrams, both parameterised by a (non-nested, separated) !-box δ_1 (among other parameters) we wish to remove our dependence on δ_1 by instead verifying a finite set of equations, each of which has a different value for δ_1. Note that for this to be the case we require the number of inputs to be equal; i.e. $m := m_1 = m_2$ and $p := p_1 = p_2$, but we do not require $n_1 = n_2$ in (8.117). We instantiate $\delta_1 = d$ and consider D_1 and D_2 to be in series !-box form, with matrix interpretations:

$$[\![\, C_1 \,]\!]\, [\![\, B_{1,q_d} \,]\!] \ldots [\![\, B_{1,q_1} \,]\!]\, [\![\, G_{1,q_0} \,]\!] \tag{8.129}$$

$$[\![\, C_2 \,]\!]\, [\![\, B_{2,q_d} \,]\!] \ldots [\![\, B_{2,q_1} \,]\!]\, [\![\, G_{2,q_0} \,]\!] \tag{8.130}$$

And we wish to know when these two interpretations are equal. Rather than consider the matrices acting on two independent spaces we view them as acting on the direct sum of those two spaces and represent these maps as block matrices. (We drop the $[\![\, \cdot \,]\!]$ notation when it would appear inside a matrix.)

$$[\![\, C_1 \,]\!]\, [\![\, B_{1,q_d} \,]\!] \ldots [\![\, B_{1,q_1} \,]\!]\, [\![\, G_{1,q_0} \,]\!] = [\![\, C_2 \,]\!]\, [\![\, B_{2,q_d} \,]\!] \ldots [\![\, B_{2,q_1} \,]\!]\, [\![\, G_{2,q_0} \,]\!] \tag{8.131}$$

$$\iff \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \ldots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 \tag{8.132}$$

One can think of the above as copying an input vector $x \in \mathbb{H}^m$ as $x :: x$ in $\mathbb{H}^m \oplus \mathbb{H}^m$, then applying

$$[\![\, C_1 \,]\!]\, [\![\, B_{1,q_d} \,]\!] \ldots [\![\, B_{1,q_1} \,]\!]\, [\![\, G_{1,q_0} \,]\!] \tag{8.133}$$

and

$$[\![\, C_2 \,]\!]\, [\![\, B_{2,q_d} \,]\!] \ldots [\![\, B_{2,q_1} \,]\!]\, [\![\, G_{2,q_0} \,]\!] \tag{8.134}$$

to the left and right copies respectively. After that we apply a minus sign to the right hand result and add that to the left hand result, effectively comparing them and demanding

the difference to be 0. We seek to prove:

$$
\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0
$$
$$
\forall d \leq N, q
$$
$$
\implies \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0
$$
$$
\forall d, q \tag{8.135}
$$

Recalling that q is the choice of values for α_j, $\delta_{k>1}$ and $v \in \mathbb{H}^{\otimes p}$, we use Q to denote the set of all possible choices. We use $\hat{B}_q$ for the matrix that acts as the direct sum of $B_{1,q}$ and $B_{2,q}$:

$$
\hat{B}_q := \begin{bmatrix} B_{1,q} & 0 \\ 0 & B_{2,q} \end{bmatrix} \tag{8.136}
$$

and inductively define the spaces:

$$
V_0 := \mathrm{span}\left\{ \bigcup_{q \in Q} \mathrm{Im}\left(\begin{bmatrix} G_{1,q} & 0 \\ 0 & G_{2,q} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} \right) \right\} \tag{8.137}
$$

$$
V_j := \mathrm{span}\left\{ V_{j-1} \cup \bigcup_{q \in Q} \hat{B}_q V_{j-1} \right\} \tag{8.138}
$$

The V_j form an ascending sequence of subspaces, each containing the potential images of up to j applications of $\hat{B}_q$:

$$
V_j \geq \mathrm{Im}(\ \hat{B}_{q_k} \dots \hat{B}_{q_1} V_0\ ) \quad \forall k \leq j\ \forall q_k, \dots, q_1 \in Q \tag{8.139}
$$

By Lemma 8.49 there is a V_b with the following properties:

- if $j < b$ then $V_j > V_{j-1}$

- if $j \geq b$ then $V_j = V_{j-1}$

- $b \leq \dim(\ \mathbb{H}^{n_1} \oplus \mathbb{H}^{n_2}\ )$

Consider the subspace W defined as

$$
W := \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \tag{8.140}
$$

We wish to show that:

$$V_j \le W \qquad\qquad \forall j \le N \qquad\qquad (8.141)$$
$$\implies V_j \le W \qquad\qquad \forall j$$

Since $V_c = V_N$ when $c \ge N \ge b$ it is enough to show that this is the case for all V_j when $j \le N$. This is, however, the assumption in our theorem; that for $d \le N$ our diagrammatic equation holds, and so for any choice of d and $q_1, \ldots, q_d$ this matrix equation holds:

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 \quad (8.142)$$

We have shown that for any choice of inputs $v \in \mathbb{H}^p \otimes \ldots \otimes \mathbb{H}^p \otimes \mathbb{H}^m$, and parameters α_j, $\delta_{k>1}$ our matrix equations hold, and by extension they hold on all elements of the space $\mathbb{H}^{\otimes(m+dp)}$, i.e. that:

$$\forall d \le N \; \forall q_d \ldots q_1$$
$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0$$
$$(8.143)$$

$$\implies \forall d \; \forall q_d \ldots q_1$$
$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0$$
$$(8.144)$$

And therefore:

$$[\![\, C_1 \,]\!]\,[\![\, B_{1,q_d} \,]\!] \ldots [\![\, B_{1,q_1} \,]\!]\,[\![\, G_{1,q_0} \,]\!] = [\![\, C_2 \,]\!]\,[\![\, B_{2,q_d} \,]\!] \ldots [\![\, B_{2,q_1} \,]\!]\,[\![\, G_{2,q_0} \,]\!]$$
$$\forall d \le N \; \forall q_d \ldots q_1$$
$$\implies [\![\, C_1 \,]\!]\,[\![\, B_{1,q_d} \,]\!] \ldots [\![\, B_{1,q_1} \,]\!]\,[\![\, G_{1,q_0} \,]\!] = [\![\, C_2 \,]\!]\,[\![\, B_{2,q_d} \,]\!] \ldots [\![\, B_{2,q_1} \,]\!]\,[\![\, G_{2,q_0} \,]\!] \quad (8.145)$$
$$\forall d \; \forall q_d \ldots q_1$$

And therefore for any choice of value for α_j and $\delta_{k>1}$:

$$\left[\!\left[\, \{\quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \delta_1 = d} \,\right]\!\right] = \left[\!\left[\, \{\quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \delta_1 = d} \,\right]\!\right] \qquad \forall d \le N$$
$$(8.146)$$

$$\implies \left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d} \right]\!\right] \qquad \forall d$$

(8.147)

$\square$

Example 8.52 (!-box verification). Consider the following Universal ZX family of equations, parameterised by a single !-box:

$$\mathbb{E} := \left\{ \quad \quad = \quad \quad \right\}_\delta$$

(8.148)

The join between the !-box and the rest of the diagram on the left is two wires (dimension 2^2), and on the right is one wire (dimension 2^1). These sum to have dimension 6, and we therefore need only to check the !-box instances $(0,\ldots,6)$ to be sure that the equation $\mathbb{E}$ is sound for any value of δ.

A longer example is provided in §E.

8.5   Verifying !-boxes and phase variables together

Theorem 8.11 and theorem 8.51 deal with equations containing multiple phase variables and nested !-boxes respectively. This section will put together the necessary results such that we can combine these approaches to deal with equations containing multiple !-boxes and phase variables, any of which could potentially be nested inside other !-boxes. For our running ZH example the previous sections covered the faces of the commutative cube, and this is the section where we put them all together.

Proposition 8.53. *Given a parameterised equation $\mathbb{E}$ and ways of finding*

- *finite verifying sets A_j for phase variables in diagrams without !-boxes (e.g. Theorem 8.11)*

- *finite verifying sets D_k for the !-boxes (e.g. Theorem 8.51)*

we may verify the entire family $\mathbb{E}$ by checking the (finite) set given by the cartesian product of all the A_j and D_k.

Proof. We define:

$$\bar{D} := D_1 \times D_2 \times \cdots \times D_m \tag{8.149}$$

$$\bar{\delta} := (\delta_1,\ldots,\delta_m) \tag{8.150}$$

$$A_j^{\bar{\delta}} := \text{The verifying set for } \alpha_j \text{ once } E \text{ has had !-boxes instantiated at } \bar{\delta} \tag{8.151}$$

Construct A_j by choosing as many points as there are in $\max_{\bar{\delta}} \left\{ |A_j^{\bar{\delta}}| \right\}$. This is finite because the D_k and the $A_j^{\bar{\delta}}$ are finite. A_j therefore contains enough points to be a verifying set for $A_j^{\bar{\delta}}$ $\forall \bar{\delta} \in \bar{D}$. We now show that $A_1 \times \cdots \times A_n \times D_1 \times \cdots \times D_m$ describes a verifying set for the parameterised equation $\mathbb{E}$:

$$
\begin{aligned}
&\mathbb{E}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \forall j,k \ \alpha_j \in A_j, \delta_k \in D_k} && \text{is sound} && \text{(8.152)} \\
=\ &\mathbb{E}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \forall i \ \alpha_j \in A_j, \bar{\delta} \in \bar{D}} && \text{is sound} && \text{(rewrite using } \bar{\delta} \text{ notation)} && \text{(8.153)} \\
\implies\ &\mathbb{E}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \forall i \ \alpha_j \in A_j^{\bar{\delta}}, \bar{\delta} \in \bar{D}} && \text{is sound} && \text{(construction of } A_j \text{)} && \text{(8.154)} \\
\implies\ &\mathbb{E}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \bar{\delta} \in \bar{D}} && \text{is sound} && \text{(theorem 8.11)} && \text{(8.155)} \\
\implies\ &\mathbb{E}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m} && \text{is sound} && \text{(theorem 8.51)} && \text{(8.156)}
\end{aligned}
$$

$\square$

We now proceed towards the main theorem of this chapter, giving the definition that we will need in the proof. We shall need algorithms to determine the sets A_j and D_k, as required by Proposition 8.53, and for that we shall need a way of ordering our !-boxes, and a notion of 'the largest diagram'. Proof of existence, or correctness, of these definitions are given in the proof of Theorem 8.58.

Definition 8.54 (!-box ordering)**.** Given a parameterised equation $\mathbb{E}$, we construct the ordered list $\delta_{k_1} \succ \delta_{k_2} \succ \cdots \succ \delta_{k_n}$ of !-boxes by recursively picking a !-box that is nested in no other !-boxes, then removing that !-box from the nesting diagram of $\mathbb{E}$. Repeat on the new nesting diagram.

Definition 8.55 (The algorithm !-Removal)**.** Given a verifying set of equations $\{\mathbb{E}_\kappa\}_{\kappa \in K}$, and a !-box δ_k nested in no other !-boxes present in the $\mathbb{E}_\kappa$:

We define !-Remove(δ_k) as the process described in Theorem 8.51. It acts on the set $\{\mathbb{E}_\kappa\}_{\kappa \in K}$ by acting on each of the $\mathbb{E}_\kappa$ in turn, finding the value N_κ, and creating the new verification set:

$$
\{\mathbb{E}_\kappa\}_{\kappa \in K'} := \bigcup_{\kappa \in K} \left\{ \mathbb{E}_{\kappa | \delta_k = 1}, \ldots, \mathbb{E}_{\kappa | \delta_k = N_\kappa} \right\} \tag{8.157}
$$

Definition 8.56 (Largest diagram)**.** Given a verifying set of equations $\{\mathbb{E}_\kappa\}_{\kappa \in K}$, we use $\bar{\mathbb{E}}$ to denote the !-box free equation that is the result of instantiating each !-box δ_k at its largest required amount N_k given by Definition 8.55

Definition 8.57 (The algorithm α-Removal)**.** Given a verifying set $\{\mathbb{E}_\kappa\}_{\kappa \in K}$ we construct the set A_j for the variable α_j by considering the degree of α_j in $\bar{\mathbb{E}}$, and then choosing enough valid values of α_j to reach the amount dictated by Theorem 8.11. We then form:

$$\{\mathbb{E}_\kappa\}_{\kappa \in K'} := \bigcup_{\kappa \in K, a \in A_j} \{\mathbb{E}_\kappa | \alpha_j = a\} \tag{8.158}$$

If we are working with $\text{RING}_\mathbb{C}$, $\text{ZW}_\mathbb{C}$, or $\text{ZH}_\mathbb{C}$ we may use Theorem 8.33 to generate the sets $A_1 := \{a_1\}, \ldots A_n := \{a_n\}$ instead.

Theorem 8.58 (Finite verification of !-boxes and phase variables)**.** *Given a parameterised family of equations* $\mathbb{E}$, *where the !-boxes are separated and well nested, we can construct a finite set of simple equations* $\{\mathbb{E}_\kappa\}_{\kappa \in K}$ *that verifies* $\mathbb{E}$.

The idea of the proof is to iteratively remove dependencies on !-boxes via Theorem 8.51, each time generating a larger set of verifying equations. Once we have removed all !-box dependence we then use the method of Proposition 8.53 to remove phase variable dependence; using the 'largest' equation in $\{\mathbb{E}_\kappa\}$ to determine the sizes of the A_j. Since every step removes either a !-box or a phase variable (and introduces neither !-boxes nor phase variables) this process terminates.

Proof. Throughout this proof we iterate on the set $\{\mathbb{E}_\kappa\}$. While at all times $\{\mathbb{E}_\kappa\}$ is a finite verifying set for $\mathbb{E}$, it is only at the end that $\{\mathbb{E}_\kappa\}$ is a finite verifying set of simple equations. We first show the existence of an ordered list of the !-boxes present in E, compatible with the nesting order on both of the nesting diagrams of E.

Claims:

1. !-Remove(δ_k) removes any dependency on δ_k in the verifying set $\{\mathbb{E}_\kappa\}_{\kappa \in K'}$

2. $\{\mathbb{E}_\kappa\}_{\kappa \in K'}$ verifies $\{\mathbb{E}_\kappa\}_{\kappa \in K}$

3. !-Remove(δ_k) does not alter the nesting ordering of any remaining !-boxes and phase variables in the verification pair

4. The ordered list $\delta_{k_1} \succ \delta_{k_2} \succ \ldots$ provides us with a sequence of !-boxes such that we can apply !-Remove($\delta_{k_{n+1}}$) to the output of !-Remove(δ_{k_n}).

5. Applying !-Remove to this ordered sequence results in a finite verifying set that has no dependence on any !-box.

Proof of claims: The first, second and fifth claims follow from Theorem 8.51. The third and fourth claims are clear from the definitions.

Starting with $\{\mathbb{E}_\kappa\}_{\kappa \in K} = \mathbb{E}$ we iteratively apply !-Remove according to the nesting order to obtain $\{\mathbb{E}_\kappa\}_{\kappa \in K'}$ which is a set of equations with phase variables but no !-boxes, and also gives us the value of $\bar{\bar{\mathbb{E}}}$ (see Definition 8.56). We then iteratively apply α-Remove to construct a finite set $\{\mathbb{E}_\kappa\}_{\kappa \in K''}$ of simple equations that verifies $\mathbb{E}$.

<div align="right">□</div>

Remark 8.59. When we introduced conjecture synthesis we said in Remark 3.2 that because we can now verify conjectures that contain !-boxes it is now worth generating conjectures that contain !-boxes. This can be done in one of two ways: We can infer !-boxes onto an existing theorem, by perhaps spotting repeating subdiagrams on both sides of the equation. Alternatively we could synthesise separated pattern graphs directly (i.e. generate diagrams with !-boxes in) and then store the first $0, \ldots, N$ instances of each !-box expansion where N is the join. This would not give us all the details needed for verifying whether two such diagrams were equal, as additional instances will need to be calculated, but would allow some initial checks to be performed quickly.

8.6   Summary

This chapter combines two very different verification results; one for phase variables (reliant on the structure of the phase algebra), and the other for !-boxes (reliant on the new series !-box form). We showed how to unify these two results into one that works in all cases where the !-boxes are separated, and also how to separate out !-boxes in most, but not all, cases. When considering qubits we used some Galois theory to reduce the number of verifying equations needed from 'finite' to 'one'. Further work includes finding similar results for the calculus ZQ, and potentially reducing the number of equations needed to verify !-boxes over qubits. For conjecture synthesis this chapter shows how to verify the sorts of hypotheses created in §7, but also shows that we can compare diagrams containing !-boxes directly if we were to generate them as part of conjecture synthesis. This chapter also gave the final piece of our conjecture inference; hypothesising theorems involving !-boxes. It also brings the results chapters of this thesis to a close.

Chapter 9

Conclusion

This thesis demonstrates that phase algebras conceal a wealth of algebraic complexity that can and should be used in both quantum computing and conjecture synthesis. This wealth comes from the interplay between the phase algebra of the diagrams and the underlying category of the interpretation. In our exploration of this theme we have introduced new graphical calculi, demonstrated a deep relationship between diagrammatic rules and algebraic geometry, and introduced phase homomorphisms to link the algebra of labels with the algebra of interpretations. While doing this we improved on the design of conjecture synthesis, exhibited new methods for conjecture verification, and combined all these, along with some Galois Theory, to show how a single equation can imply an abundance of generalisations.

The impetus for this work was the task of Conjecture Synthesis for Quantum Graphical Calculi. We have extended the core process found in the literature by including a generalisation step at the point of successful synthesis. This generalisation step is performed (although not exclusively) via the discovery, presented in this thesis, of a link between our rule presentations and algebraic geometry. Through the related work on conjecture verification, also presented in this thesis, it was shown that the evidence needed for such a generalisation could be as low as a single equation; this is equivalent to inferring all the points of a surface based on just a single point and some symmetries.

These symmetries of the parameter space, also discovered in this thesis, are the phase homomorphism pairs exhibited in §5. We introduced them, demonstrated when they preserved soundness and derivations, and then classified them for ZW, ZH, RING, and certain fragments of ZX. The author hopes that this examination, and the contents of §5 overall, prompt further research into graphical calculi with a greater variety of phase algebras. These parameter spaces do not arise by chance: They are a direct consequence of our choices of phase algebras, and have the capacity to reveal profound insights into our graphical calculi.

The conjecture verification results of this thesis rely on the (relatively simple) properties of polynomial interpolation and chains of vector subspaces. The hard work in

these results was in extracting the necessary information from the diagrams. Again we emphasise that this algebraic machinery is present, but because we are dealing with an unusual presentation of the information, and in a context where these results are less commonly applied, these properties are far easier to miss.

Graphical calculus design is therefore clearly important but we have also shown ways in which it is (usefully) limited. Any future work to design a quantum graphical calculus with a phase ring will necessarily construct something similar to RING$_\mathbb{C}$ (Remark 5.36). This is not to say that RING$_\mathbb{C}$ is the only option (ZH and ZW have good reasons for their choices of generators) but that RING$_\mathbb{C}$ will act as common ground between these calculi, and will make the construction of equivalences easier. The language ZQ, on the other hand, will be useful for those creating optimisers, or reasoning about certain hardware implementations. In fact the author would argue that the rules of ZQ are more intuitive than those for Universal ZX, and so there is a pedagogical argument for explaining quantum circuitry using ZQ. The unwieldy nature of the Euler Decomposition rule of ZX, and its comparative elegance in ZQ, was a major influence on the direction of this research. Both ZQ and RING were introduced and then showed to be sound and complete in this thesis.

The central point of this thesis was that phase algebras are important, and that they enable algebraic machinery that was previously underused. This is a continuation of the program of Categorical Quantum Mechanics, a program that aims to make reasoning about these systems easier through selective abstraction:

> The arguments for the benefits of a high-level, conceptual approach to designing and reasoning about quantum computational systems are just as compelling as for classical computation. In particular, we have in mind the hard-learned lessons from Computer Science of the importance of compositionality, types, abstraction, and the use of tools from algebra and logic in the design and analysis of complex informatic processes. (Categorical Quantum Mechanics [AC09])

As a final demonstration of this point let us examine the generalisation step of an updated simple RING$_\mathbb{C}$ conjecture synthesis run:

- First an unparameterised, sound equation in RING$_\mathbb{C}$ is generated.

- Let us assume that this equation shares a skeleton with other equations, already shown to be sound. We may then use the link between the algebra of equations and the geometry of parameter spaces to infer a potential generalisation.

- We can then use the results of §8.3 to pick a single, unparameterised, verifying equation for this generalisation. This equation will already be in our search space, we simply move it forward in the queue.

- With our generalisation verified we can then reduce the search space by a far greater amount than if we had not performed the verification.

This generalisation process uses Algebraic Geometry, Galois Theory, Laurent Polynomials, Polynomial Interpolation, and Category Theory. Each of their particular uses are novel to this thesis. All of these uses rely on the interplay between the phase algebra and the underlying matrices.

The author hopes that this research prompts further work in the program of Categorical Quantum Mechanics, in particular further collaboration between quantum computing and traditionally algebraic fields. The link to algebraic geometry given here deserves further exploration, both for quantum computing and for the inference step in conjecture synthesis. Even outside the geometric inference framework of this thesis our novel generalisation step should increase efficacy in future conjecture synthesis projects. The calculus ZQ would be ideal for formal verification of optimisers such as TriQ.

Acknowledgements

Bibliography

[AC09]   Samson Abramsky and Bob Coecke. 'Categorical Quantum Mechanics'. In: *Handbook of Quantum Logic and Quantum Structures* (2009), pp. 261–323. DOI: 10.1016/b978-0-444-52869-8.50010-4.

[Amm93]   Kurt Ammon. 'A learning procedure for mathematics'. In: *Annals of Mathematics and Artificial Intelligence* 8.3 (Sept. 1993), pp. 407–423. ISSN: 1573-7470. DOI: 10.1007/BF01530800.

[Art13]   Michael Artin. *Algebra: Pearson New International Edition*. Pearson Education Limited, 2013. ISBN: 1292027665.

[Aru+19]   Frank Arute et al. 'Quantum supremacy using a programmable superconducting processor'. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5.

[Bac+20]   Miriam Backens et al. *There and back again: A circuit extraction tale*. 2020. arXiv: 2003.01664 [quant-ph].

[Bac14a]   Miriam Backens. 'The ZX-calculus is complete for stabilizer quantum mechanics'. In: *New Journal of Physics* 16.9 (Sept. 2014), p. 093021. ISSN: 1367-2630. DOI: 10.1088/1367-2630/16/9/093021.

[Bac14b]   Miriam Backens. 'The ZX-calculus is complete for the single-qubit Clifford+T group'. In: *Electronic Proceedings in Theoretical Computer Science* 172 (Dec. 2014), pp. 293–303. ISSN: 2075-2180. DOI: 10.4204/eptcs.172.21.

[Bac15]   Miriam Backens. 'Making the stabilizer ZX-calculus complete for scalars'. In: *Proceedings of the 12th International Workshop on Quantum Physics and Logic (QPL 2015)*. Ed. by Chris Heunen, Peter Selinger and Jamie Vicary. Vol. 195. Electronic Proceedings in Theoretical Computer Science. 2015, pp. 17–32. DOI: 10.4204/EPTCS.195.2.

[Bac16]   Miriam Backens. 'Completeness and the ZX-calculus'. PhD thesis. University of Oxford, 2016. eprint: arXiv:1602.08954.

[BBW20]   Niel de Beaudrap, Xiaoning Bian and Quanlong Wang. 'Techniques to Reduce $\pi/4$-Parity-Phase Circuits, Motivated by the ZX Calculus'. In: *Electronic Proceedings in Theoretical Computer Science* 318 (May 2020), pp. 131–149. ISSN: 2075-2180. DOI: `10.4204/eptcs.318.9`.

[BCP97]   Wieb Bosma, John Cannon and Catherine Playoust. 'The Magma algebra system. I. The user language'. In: *J. Symbolic Comput.* 24.3-4 (1997). Computational algebra and number theory (London, 1993), pp. 235–265. ISSN: 0747-7171. DOI: `10.1006/jsco.1996.0125`.

[BD15]   Miriam Backens and Ali Nabi Duman. 'A Complete Graphical Calculus for Spekkens' Toy Bit Theory'. In: *Foundations of Physics* 46.1 (Oct. 2015), pp. 70–103. ISSN: 1572-9516. DOI: `10.1007/s10701-015-9957-7`.

[BH20]   Niel de Beaudrap and Dominic Horsman. 'The ZX calculus is a language for surface code lattice surgery'. In: *Quantum* 4 (2020). DOI: `10.22331/q-2020-01-09-218`.

[BK19]   Miriam Backens and Aleks Kissinger. 'ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity'. In: *Electronic Proceedings in Theoretical Computer Science* 287 (Jan. 2019), pp. 23–42. ISSN: 2075-2180. DOI: `10.4204/eptcs.287.2`.

[BN98]   Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* New York, NY, USA: Cambridge University Press, 1998. ISBN: 0-521-45520-0.

[BPW17]   Miriam Backens, Simon Perdrix and Quanlong Wang. 'A Simplified Stabilizer ZX-calculus'. In: *Proceedings 13th International Conference on Quantum Physics and Logic, Glasgow, Scotland, 6-10 June 2016*. Ed. by Ross Duncan and Chris Heunen. Vol. 236. Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, 2017, pp. 1–20. DOI: `10.4204/EPTCS.236.1`.

[CD08]   Bob Coecke and Ross Duncan. 'Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II'. In: ed. by Luca Aceto et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Chap. Interacting Quantum Observables, pp. 298–310. ISBN: 978-3-540-70583-3. DOI: `10.1007/978-3-540-70583-3_25`.

[CE11]   Bob Coecke and Bill Edwards. 'Three qubit entanglement within graphical Z/X-calculus'. In: *Electronic Proceedings in Theoretical Computer Science* 52 (Mar. 2011), pp. 22–33. ISSN: 2075-2180. DOI: `10.4204/eptcs.52.3`.

[CHP19]   Titouan Carette, Dominic Horsman and Simon Perdrix. 'SZX-Calculus: Scalable Graphical Quantum Reasoning'. In: *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Ed. by Peter Rossmanith, Pinar Heggernes and Joost-Pieter Katoen. Vol. 138. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 55:1–55:15. ISBN: 978-3-95977-117-7. DOI: 10.4230/LIPIcs.MFCS.2019.55.

[CK17]   B. Coecke and A. Kissinger. *Picturing Quantum Processes*. Cambridge University Press, 2017. ISBN: 9781107104228.

[Coe+11]   Bob Coecke et al. 'The GHZ/W-calculus contains rational arithmetic'. In: *Electronic Proceedings in Theoretical Computer Science* 52 (Mar. 2011), pp. 34–48. ISSN: 2075-2180. DOI: 10.4204/eptcs.52.4.

[Cow+20]   Alexander Cowtan et al. 'Phase Gadget Synthesis for Shallow Circuits'. In: *Electronic Proceedings in Theoretical Computer Science* 318 (May 2020), pp. 213–228. ISSN: 2075-2180. DOI: 10.4204/eptcs.318.13.

[CQC19]   *Cambridge Quantum Computing publications webpage*. https://cambridgequantum.com/publications/. Accessed: 2020-01-23. 2019.

[CR79]   Charles J Colbourn and Ronald C Read. 'Orderly algorithms for generating restricted classes of graphs'. In: *Journal of Graph Theory* 3.2 (1979), pp. 187–195. ISSN: 1097-0118. DOI: 10.1002/jgt.3190030210.

[DK13]   LUCAS DIXON and ALEKS KISSINGER. 'Open-graphs and monoidal theories'. In: *Mathematical Structures in Computer Science* 23.02 (Feb. 2013), pp. 308–359. ISSN: 1469-8072. DOI: 10.1017/s0960129512000138.

[DP09]   Ross Duncan and Simon Perdrix. 'Graph States and the Necessity of Euler Decomposition'. In: *Mathematical Theory and Computational Practice*. Ed. by Klaus Ambos-Spies, Benedikt Löwe and Wolfgang Merkle. Vol. 5635. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 167–177. ISBN: 978-3-642-03072-7. DOI: 10.1007/978-3-642-03073-4_18.

[DP14]   Ross Duncan and Simon Perdrix. 'Pivoting makes the ZX-calculus complete for real stabilizers'. In: *Electronic Proceedings in Theoretical Computer Science* 171 (Dec. 2014), pp. 50–62. ISSN: 2075-2180. DOI: 10.4204/eptcs.171.5.

[Dun+20]   Ross Duncan et al. 'Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus'. In: *Quantum* 4 (June 2020), p. 279. ISSN: 2521-327X. DOI: 10.22331/q-2020-06-04-279.

[EPS73]   Hartmut Ehrig, Michael Pfender and Hans Jürgen Schneider. 'Graph-grammars: An algebraic approach'. In: *14th Annual Symposium on Switching and Automata Theory (swat 1973)*. IEEE. 1973, pp. 167–180. DOI: 10.1109/SWAT.1973.11.

[GF19]   Craig Gidney and Austin G. Fowler. 'Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation'. In: *Quantum* 3 (Apr. 2019), p. 135. ISSN: 2521-327X. DOI: `10.22331/q-2019-04-30-135`.

[Got98]   Daniel Gottesman. *The Heisenberg Representation of Quantum Computers.* 1998. arXiv: `quant-ph/9807006 [quant-ph]`.

[GS00]   Mariano Gasca and Thomas Sauer. 'Polynomial interpolation in several variables'. In: *Advances in Computational Mathematics* 12.4 (Mar. 2000), p. 377. ISSN: 1572-9044. DOI: `10.1023/A:1018981505752`.

[Had15]   Amar Hadzihasanovic. 'A Diagrammatic Axiomatisation for Qubit Entanglement'. In: *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science* (July 2015). DOI: `10.1109/lics.2015.59`.

[Had17]   A. Hadzihasanovic. 'The algebra of entanglement and the geometry of composition'. PhD thesis. University of Oxford, Sept. 2017. arXiv: `1709.08086 [math.CT]`.

[HGK04]   Michiel Hazewinkel, Nadiya Gubareni and Vladimir V Kirichenko. *Algebras, rings and modules.* Vol. 1. Springer Science & Business Media, 2004. DOI: `10.1007/1-4020-2691-9`.

[Hum03]   David Hume. *A treatise of human nature.* Dover Publications, 2003. ISBN: 0486432505.

[IBM19]   International Business Machines Corporation. *On "Quantum Supremacy".* `https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/`. Accessed: 2020-01-21. 2019.

[JDB09]   Moa Johansson, Lucas Dixon and Alan Bundy. 'IsaCoSy: Synthesis of inductive theorems'. In: *Workshop on Automated Mathematical Theory Exploration (Automatheo).* 2009. URL: `http://hdl.handle.net/1842/4751`.

[JDB11]   Moa Johansson, Lucas Dixon and Alan Bundy. 'Conjecture Synthesis for Inductive Theories'. In: *Journal of Automated Reasoning* 47.3 (Oct. 2011), pp. 251–289. ISSN: 1573-0670. DOI: `10.1007/s10817-010-9193-y`.

[Jea+17]   Emmanuel Jeandel et al. 'ZX-Calculus: Cyclotomic Supplementarity and Incompleteness for Clifford+T Quantum Mechanics'. In: *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017).* Ed. by Kim G. Larsen, Hans L. Bodlaender and Jean-Francois Raskin. Vol. 83. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 11:1–11:13. ISBN: 978-3-95977-046-0. DOI: `10.4230/LIPIcs.MFCS.2017.11`.

[JPV18a]   Emmanuel Jeandel, Simon Perdrix and Renaud Vilmart. 'A Complete Ax-
           iomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics'. In:
           *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Com-
           puter Science*. LICS '18. Oxford, United Kingdom: ACM, 2018, pp. 559–568.
           ISBN: 978-1-4503-5583-4. DOI: `10.1145/3209108.3209131`.

[JPV18b]   Emmanuel Jeandel, Simon Perdrix and Renaud Vilmart. 'Diagrammatic
           Reasoning beyond Clifford+T Quantum Mechanics'. In: *Proceedings of the
           33rd Annual ACM/IEEE Symposium on Logic in Computer Science - LICS
           '18* (2018). DOI: `10.1145/3209108.3209139`.

[JPV19]   Emmanuel Jeandel, Simon Perdrix and Renaud Vilmart. 'A Generic Normal
           Form for ZX-Diagrams and Application to the Rational Angle Complete-
           ness'. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer
           Science (LICS)* (June 2019). DOI: `10.1109/lics.2019.8785754`.

[JS91]   André Joyal and Ross Street. 'The geometry of tensor calculus, I'. In: *Ad-
           vances in mathematics* 88.1 (1991), pp. 55–112. DOI: `10.1016/0001-8708(91)
           90003-P`.

[Kis12]   Aleks Kissinger. 'Synthesising Graphical Theories'. PhD thesis. University
           of Oxford, 2012. arXiv: `1202.6079`.

[Kis13]   Aleks Kissinger. 'Synthesising Graphical Theories'. In: *ATx'12/WInG'12:
           Joint Proceedings of the Workshops on Automated Theory eXploration and
           on Invariant Generation*. Ed. by Jacques Fleuriot et al. Vol. 17. EPiC Series
           in Computing. EasyChair, 2013, pp. 26–35. DOI: `10.29007/5dkd`.

[KMS14]   Aleks Kissinger, Alex Merry and Matvey Soloviev. 'Pattern graph rewrite
           systems'. In: vol. 143. Open Publishing Association, Mar. 2014, pp. 54–66.
           DOI: `10.4204/eptcs.143.5`.

[KW20a]   Aleks Kissinger and John van de Wetering. 'PyZX: Large Scale Automated
           Diagrammatic Reasoning'. In: *Electronic Proceedings in Theoretical Com-
           puter Science* 318 (May 2020), pp. 229–241. ISSN: 2075-2180. DOI: `10.4204/
           eptcs.318.14`.

[KW20b]   Aleks Kissinger and John van de Wetering. 'Reducing the number of non-
           Clifford gates in quantum circuits'. In: *Physical Review A* 102.2 (Aug. 2020).
           ISSN: 2469-9934. DOI: `10.1103/physreva.102.022406`.

[KZ15]   Aleks Kissinger and Vladimir Zamdzhiev. 'Quantomatic: A Proof Assistant
           for Diagrammatic Reasoning'. In: *Lecture Notes in Computer Science* (2015),
           pp. 326–336. ISSN: 1611-3349. DOI: `10.1007/978-3-319-21401-6_22`.

[Mac13]   Saunders Mac Lane. *Categories for the working mathematician*. Vol. 5. Springer
           Science & Business Media, 2013. DOI: `10.1007/978-1-4757-4721-8`.

[Mac65]   Saunders MacLane. 'Categorical algebra'. In: *Bulletin of the American Mathematical Society* 71.1 (Jan. 1965), pp. 40–106. DOI: `10.1090/S0002-9904-1965-11234-4`.

[Mar00]   Thomas Sauer Mariano Gasca. 'On the history of multivariate polynomial interpolation.' In: *Journal of Computational and Applied Mathematics* 122 (2000), pp. 23–35. DOI: `https://doi.org/10.1016/S0377-0427(00)00353-8`.

[Mer14]   Alexander Merry. 'Reasoning with !-Graphs'. PhD thesis. University of Oxford, 2014. arXiv: `1403.7828 [cs.LO]`.

[Mil19]   Hector Miller-Bakewell. *ZX-calculus and accessibility*. `http://zxcalculus.com/accessibility.html`. Accessed: 2020-01-23. 2019.

[Mil20]   Hector Miller-Bakewell. 'Finite Verification of Infinite Families of Diagram Equations'. In: *Electronic Proceedings in Theoretical Computer Science* 318 (May 2020), pp. 27–52. ISSN: 2075-2180. DOI: `10.4204/eptcs.318.3`.

[Mur+19]   Prakash Murali et al. 'Full-Stack, Real-System Quantum Computer Studies: Architectural Comparisons and Design Insights'. In: *Proceedings of the 46th International Symposium on Computer Architecture*. ISCA '19. Phoenix, Arizona: Association for Computing Machinery, 2019, pp. 527–540. ISBN: 9781450366694. DOI: `10.1145/3307650.3322273`.

[NC10]   Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010. DOI: `10.1017/CBO9780511976667`.

[NW17]   Kang Feng Ng and Quanlong Wang. *A universal completion of the ZX-calculus*. 2017. arXiv: `1706.09877 [quant-ph]`.

[Pen71]   Roger Penrose. 'Applications of negative dimensional tensors'. In: *Combinatorial mathematics and its applications* 1 (1971), pp. 221–244.

[Per+12]   Alejandro Perdomo-Ortiz et al. 'Finding low-energy conformations of lattice protein models by quantum annealing'. In: *Scientific Reports* 2.1 (Aug. 2012), p. 571. ISSN: 2045-2322. DOI: `10.1038/srep00571`.

[Pre18]   John Preskill. 'Quantum Computing in the NISQ era and beyond'. In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: `10.22331/q-2018-08-06-79`.

[Qui15]   David Quick. '!-Logic : first order reasoning for families of non-commutative string diagrams'. PhD thesis. University of Oxford, 2015. URL: `https://ora.ox.ac.uk/objects/uuid:baf2d50d-8c5f-419d-9b3d-f2f700f8acbd`.

[Ran14]   André Ranchin. 'Depicting qudit quantum mechanics and mutually unbiased qudit theories'. In: *Electronic Proceedings in Theoretical Computer Science* 172 (Dec. 2014), pp. 68–91. ISSN: 2075-2180. DOI: `10.4204/eptcs.172.6`.

[Sch13]   Christian Schröder de Witt. 'The ZX calculus is incomplete for non-stabilizer quantum mechanics'. University of Oxford Master thesis. 2013. URL: http://www.cs.ox.ac.uk/people/bob.coecke/Christian_thesis.pdf.

[Sel07]   Peter Selinger. 'Dagger Compact Closed Categories and Completely Positive Maps: (Extended Abstract)'. In: *Electronic Notes in Theoretical Computer Science* 170 (2007). Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005), pp. 139–163. ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2006.12.018.

[Sel11]   P. Selinger. 'A Survey of Graphical Languages for Monoidal Categories'. In: *New Structures for Physics*. Ed. by Bob Coecke. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 289–355. ISBN: 978-3-642-12821-9. DOI: 10.1007/978-3-642-12821-9_4.

[Sho85]   Ken Shoemake. 'Animating Rotation with Quaternion Curves'. In: *SIGGRAPH Comput. Graph.* 19.3 (July 1985), pp. 245–254. ISSN: 0097-8930. DOI: 10.1145/325165.325242.

[Sho94]   P. Shor. 'Algorithms for quantum computation: discrete logarithms and factoring'. In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[Smi+04]   Karen Smith et al. *An Invitation to Algebraic Geometry*. Springer Science & Business Media, 2004. ISBN: 978-1-4419-3195-5. DOI: 10.1007/978-1-4757-4497-2.

[Ste15]   Ian Stewart. *Galois Theory*. Vol. Fourth edition. Chapman and Hall/CRC, 2015. ISBN: 9781138401709.

[SZ14]   Christian Schröder de Witt and Vladimir Zamdzhiev. 'The ZX-calculus is incomplete for quantum mechanics'. In: *Electronic Proceedings in Theoretical Computer Science* 172 (Dec. 2014), pp. 285–292. ISSN: 2075-2180. DOI: 10.4204/eptcs.172.20.

[VBE96]   Vlatko Vedral, Adriano Barenco and Artur Ekert. 'Quantum networks for elementary arithmetic operations'. In: *Phys. Rev. A* 54 (1 July 1996), pp. 147–153. DOI: 10.1103/PhysRevA.54.147.

[Vil19]   Renaud Vilmart. 'A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics'. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (June 2019). DOI: 10.1109/lics.2019.8785765.

[Wan19]   Quanlong Wang. *An algebraic axiomatisation of ZX-calculus*. 2019. arXiv: 1911.06752 [quant-ph].

Appendices

A   Extending the calculus ZH

In this appendix:

  ▷ We extend the calculus $\mathrm{ZH}_\mathbb{C}$ to the calculus ZH_R, where R is any commutative ring with a half.

  ▷ The proof of this result is very similar to the original proof for $\mathrm{ZH}_\mathbb{C}$ in Ref. [BK19], and we imitate, quote, and make use of that paper where possible.

For the definition of the calculus ZH_R see Section 4.4.1. The majority of the text below is from Ref. [BK19], it has simply been altered to reflect the more general case of ZH_R over $\mathrm{ZH}_\mathbb{C}$. It can be seen from the proofs in Ref. [BK19] that the only property used of the ring $\mathbb{C}$ is the existence of the element $\frac{1}{2}$.

We will show that ZH_R is complete by demonstrating the existence of a unique normal form for ZH_R diagrams. It is first worth noting that, because we can turn inputs into outputs arbitrarily (since all the generators are spiders), it suffices to consider diagrams which have only outputs.

For states ψ, ϕ, let $\psi * \phi$ be the *Schur product* of ψ and ϕ obtained by plugging the i-th output of ψ and ϕ into , for each i:

$$\tag{1}$$
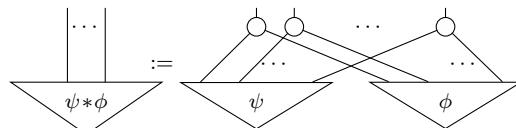
It follows from (ZS1) that $*$ is associative and commutative, so we can write k-fold Schur products $\psi_1 * \psi_2 * \ldots * \psi_k$ without ambiguity. For any finite set J with $|J| = k$, let $\prod_{j \in J} \psi_j$ be the k-fold Schur product.

Let $\mathbb{B}^n$ be the set of all n-bitstrings. For any $\vec{b} := b_1 \ldots b_n \in \mathbb{B}^n$, define the *indexing map* $\iota_{\vec{b}}$ as follows:

$$\iota_{\vec{b}} \;=\; \boxed{\vec{b}} \;=\; \left(\ominus \right)^{1-b_1} \ldots \left(\ominus \right)^{1-b_n}. \tag{2}$$
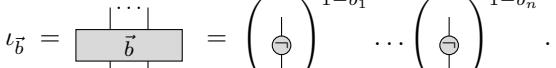
Then normal forms are given by the following 2^n-fold Schur products:

$$\prod_{\vec{b} \in \mathbb{B}^n} \left(\iota_{\vec{b}} \circ H_n(a_{\vec{b}}) \right) \tag{3}$$

193

where $H_n(a_{\vec{b}})$ is the arity-n H-box (considered as a state) labelled by an arbitrary elements $a_{\vec{b}}$ of R.

A normal form diagram can be seen as a collection of n spiders, fanning out to 2^n H-boxes, each with a distinct configuration of NOT's corresponding to the 2^n bitstrings in $\mathbb{B}^n$. Diagrammatically, normal forms are:

$$ (4) $$

Theorem A.1. *[BK19, Theorem 3.1] Normal forms are unique. In particular:*

$$\left[\!\!\left[\prod_{\vec{b}\in\mathbb{B}^n} \left(\iota_{\vec{b}} \circ H_n(a_{\vec{b}}) \right) \right]\!\!\right] = \sum_{\vec{b}\in\mathbb{B}^n} a_{\vec{b}} |\,\vec{b}\,\rangle. \tag{5}$$

Proof. The map $\iota_{\vec{b}}$ is a permutation that acts on computational basis elements as $|\,\vec{c}\,\rangle \mapsto |\,\vec{c} \oplus \vec{b} \oplus \vec{1}\,\rangle$. In particular, it sends the basis element $|\,\vec{1}\,\rangle$ to $|\,\vec{b}\,\rangle$. Hence $\iota_{\vec{b}} \circ H_n(a_{\vec{b}})$ is a vector with $a_{\vec{b}}$ in the $\vec{b}$-th component and 1 everywhere else. The Schur product of all such vectors gives the RHS of (5). $\qquad\square$

Since equation (5) gives us a means of constructing any vector in R^{2^n}, Theorem A.1 can also be seen as a proof of universality of ZH_R.

Lemma A.2 ([BK19, Lemma 3.2]). *The NOT operator copies through white spiders:*

$$ (6) $$

Lemma A.3 ([BK19, Lemma 3.3]). *The $\iota_{\vec{b}}$ operator copies through white spiders, i.e. for any $\vec{b}\in\mathbb{B}^n$:*

$$ (7) $$

Lemma A.4 ([BK19, Lemma 3.4]). *ZH_R enables the computation of the Schur product of two*

maps of the form $\iota_{\vec{b}} \circ H_n(x)$ and $\iota_{\vec{b}} \circ H_n(y)$ for any $\vec{b} \in \mathbb{B}^n$ and $x, y \in R$:

$$\tag{8}$$

We will now show that normal form diagrams, when combined in various ways, can also be put into normal form. Let

$$\tag{9}$$

denote an arbitrary normal-form diagram. It is straightforward to see that permuting the outputs of a normal-form diagram merely interchanges the bits in the coefficients $a_{\vec{b}}$. Hence, normal forms are preserved under permutations of outputs. Furthermore:

Proposition A.5 ([BK19, Proposition 3.5]). *A diagram consisting of a normal form diagram juxtaposed with* φ *can be brought into normal form using the rules of ZH_R:*

$$\tag{10}$$

Proposition A.6 ([BK19, Proposition 3.6]). *The Schur product of two normal form diagrams can be brought into normal form using the rules of ZH_R.*

$$\tag{11}$$

Corollary A.7 ([BK19, Corollary 3.7]). *The tensor product of two normal form diagrams can be brought into normal form using the rules of ZH_R.*

Remark A.8. Note that a single scalar H-box is a normal form diagram. Corollary A.7 thus implies that a diagram consisting of a normal form diagram juxtaposed with a scalar H-box can be brought into normal form. In the following proofs, we will therefore ignore scalars for simplicity: they can be added back in and then incorporated to the normal form without problems.

Proposition A.9 ([BK19, Proposition 3.9]). *The diagram resulting from applying* φ *to an output of a normal form diagram can be brought into normal form:*

$$\tag{12}$$

Our strategy will now be to show that any diagram can be decomposed into H-boxes, combined via the operations of extension, convolution, and contraction. This will give us a completeness proof, thanks to the following proposition.

195

Lemma A.10. *[[BK19, Lemma 3.10]] Any H-box can be brought into normal form using the rules of ZH_R.*

Corollary A.11 ([BK19, Corollary 3.11]). *The diagram of a single cup can be brought into normal form:*

$$\cup \quad = \quad \boxed{}_{NF} \tag{13}$$

Corollary A.12 ([BK19, Corollary 3.12]). *The diagram resulting from applying $\;\;\;$ to a pair of outputs of a normal form diagram can be brought into normal form.*

$$\boxed{NF}\cdots \quad = \quad \boxed{NF\,'}\cdots \tag{14}$$

Corollary A.13 ([BK19, Corollary 3.13]). *Applying a cap to a normal form diagram results in another normal form diagram:*

$$\boxed{NF}\cdots \quad = \quad \boxed{NF\,'}\cdots \tag{15}$$

Thanks to Corollaries A.7 and A.13, we are able to turn any diagram of normal forms into a normal form. It only remains to show that the generators of ZH_R can themselves be made into normal forms. We have already shown the result for H-boxes, so we only need the following.

Lemma A.14 ([BK19, Lemma 3.14]). *Any Z-spider can be brought into normal form using the rules of ZH_R.*

Theorem A.15 (ZH_R is complete). *For any ZH_R diagrams D_1 and D_2, if $[\![\, D_1 \,]\!] = [\![\, D_2 \,]\!]$ then D_1 is convertible into D_2 using the rules of ZH_R.*

Proof. By Theorem A.1, it suffices to show that any ZH_R diagram can be brought into normal form. Lemmas A.10 and A.14 suffice to turn any generator into normal form. Corollary A.7 lets us turn any tensor product of generators into a normal form and Corollary A.13 lets us normalise any arbitrary wiring. □

B ZH_R's relationship to RING

In this appendix we prove completeness for the set of rules for RING presented in §4.4.3, using a translation of the complete set of rules for ZH_R given in Figure 4.8. The translation was provided in Figure 4.9.

B.1 Translations of generators of RING to ZH and back

- Translation of from $\textsc{ring}_R$ to ZH and back:

$$\left[\!\left[\,\left[\!\left[\ \vphantom{X}\ \right]\!\right]_{\text{ZH}}\,\right]\!\right]_{\text{RING}} = \left[\!\left[\ \vphantom{X}\ \right]\!\right]_{\text{RING}} = \ \vphantom{X} \tag{16}$$

- Translation of from $\textsc{ring}_R$ to ZH and back:

$$\left[\!\left[\,\left[\!\left[\ \vphantom{X}\ \right]\!\right]_{\text{ZH}}\,\right]\!\right]_{\text{RING}} = \left[\!\left[\ \vphantom{X}\ \right]\!\right]_{\text{RING}} = \tag{17}$$

- Translation of from $\textsc{ring}_R$ to ZH and back:

$$\left[\!\left[\,\left[\!\left[\ \vphantom{X}\ \right]\!\right]_{\text{ZH}}\,\right]\!\right]_{\text{RING}} = \left[\!\left[\ \vphantom{X}\ \right]\!\right]_{\text{RING}} = \tag{18}$$

B.2 Translation of the derived generators of ZH

$$\tag{19}$$

B.3 Proof of completeness

In this section we prove the following:

Theorem 4.40 (RING is complete, via ZH$_R$). *The rules of §4.4.3 are complete for the graphical calculus* RING$_R$*, where R is a commutative ring with a half.*

We will show that the rules of §4.4.3, here called RING$_R^2$, can derive all of the equations $[\![\, L = R \,]\!]_{\text{RING}}$ where $L = R$ is a rule of ZH, and all the equations $[\![\, [\![\, g \,]\!]_{\text{ZH}} \,]\!]_{\text{RING}} = g$ where g is a generator of RING. First we will show some useful intermediate results.

Proposition B.1. *In* RING$_R^2$ *we can 'cycle' the gates of two adjacent Hadamard maps, and call this rule H'*

$$\tag{H'}$$

Proof. We simultaneously pre- and post-compose both sides of rule H with the elements and for the first equality, then and for the second. □

Proposition B.2. *We can also find another useful representation of the Hadamard gate, and call this rule H''*

$$\tag{H''}$$

Proof. We apply the right hand side of this rule (minus the scalar) to both sides of rule H, then cancel terms using $+_a$ and S. □

Proposition B.3. *The following derivation is helpful for the rules translated from ZH that contain* *:*

$$\tag{¬}$$

198

Proof.

$$\vcenter{\hbox{(diagram)}} \quad \overset{=}{H, sc} \quad \vcenter{\hbox{(diagram)}} \quad \overset{=}{+, S} \quad \vcenter{\hbox{(diagram)}} \tag{20}$$

$$\overset{=}{\overline{H''}} \quad \vcenter{\hbox{(diagram)}} \quad \overset{=}{\overline{\overline{D}}} \quad \vcenter{\hbox{(diagram)}} \quad \overset{=}{S, +_a} \quad \vcenter{\hbox{(diagram)}} \tag{21}$$

$$\overset{=}{\overline{\overline{D}}} \quad \vcenter{\hbox{(diagram)}} \quad \overset{=}{\overline{\overline{S}}} \quad \vcenter{\hbox{(diagram)}} \tag{22}$$

□

Corollary B.4. *The derived generator* $\boxed{a}$ *obeys the following rules that follow immediately from* $+$, $+_a$ *and* D, *so we will use these labels when referring to these rules:*

$$\vcenter{\hbox{(diagram)}} \overset{=}{+} \vcenter{\hbox{(diagram)}} \qquad \vcenter{\hbox{(diagram)}} \overset{=}{+_a} \vcenter{\hbox{(diagram)}} \qquad \vcenter{\hbox{(diagram)}} \overset{=}{D} \vcenter{\hbox{(diagram)}} \tag{23}$$

Lemma B.5. $\mathrm{RING}^2_R \vdash F(ZH_{ZS1})$, *i.e.*

$$\mathrm{RING}^2_R \vdash \quad \overbrace{\vcenter{\hbox{(diagram)}}}^{n} \quad = \quad \overbrace{\vcenter{\hbox{(diagram)}}}^{n} \tag{24}$$

Proof. This follows immediately from the rule S1 □

199

Lemma B.6. $\mathrm{RING}_R^2 \vdash F(ZH_{HS1})$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \quad \cdots \quad = \quad \cdots \tag{25}$$

Proof.

$$\overset{=}{S} \quad \overset{=}{H'} \quad \overset{=}{H} \quad \overset{=}{S} \tag{26}$$

□

Proposition B.7. $\mathrm{RING}_R^2 \vdash F(ZH_{ZS2})$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \quad = \quad = \quad \tag{27}$$

Proof.

$$\overset{=}{S} \quad \overset{=}{\times_1} \quad \tag{28}$$

□

200

Proposition B.8. $\mathrm{RING}_R^2 \vdash F(ZH_{HS2})$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \quad \begin{array}{c} \square \\ \square \end{array} \quad = \quad \bigcirc \quad \Big| \tag{29}$$

Proof. This is the same as rule H $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Proposition B.9. $\mathrm{RING}_R^2 \vdash F(ZH_{BA1})$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \qquad \qquad = \qquad \qquad \tag{30}$$

Proof. This is the same as rule BA1 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Proposition B.10. $\mathrm{RING}_R^2 \vdash F(ZH_{BA2})$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \qquad \qquad = \qquad \qquad \tag{31}$$

Proof.

$$\overset{=}{\underset{S}{=}} \qquad \overset{=}{\underset{+a}{=}} \qquad \tag{32}$$

$$\overset{=}{H} \quad \overset{=}{sc, BA2} \quad \overset{=}{+_a, S} \tag{33}$$

$\square$

Lemma B.11. $\mathrm{RING}_R^2 \vdash F(ZH_M)$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \tag{34}$$

Proof.

$$\overset{=}{\times} \quad \overset{=}{+} \tag{35}$$

$\square$

Lemma B.12. $\mathrm{RING}_R^2 \vdash F(ZH_U)$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \tag{36}$$

Proof. This is an instance of rule $+$ $\square$

202

Lemma B.13. $\mathrm{RING}_R^2 \vdash$ *equation* (17) *(the re-translation of the + generator), i.e.*

$$\mathrm{RING}_R^2 \vdash \quad \text{} \tag{37}$$

Proof.

$$\text{} \tag{38}$$

$$\overset{=}{H,sc} \quad \overset{=}{+,S} \quad \overset{=}{S} \tag{39}$$

$$\overset{=}{H'',sc} \quad \overset{=}{a} \quad \overset{=}{S} \quad \overset{=}{+'} \tag{40}$$

□

Lemma B.14. $\mathrm{RING}_R^2 \vdash F(ZH_A)$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \qquad = \tag{41}$$

204

Proof.

$$\underset{H, sc}{=} \qquad \underset{+_a, S}{=} \tag{42}$$

$$\underset{S}{=} \qquad \underset{+}{=} \tag{43}$$

$$\underset{H''}{=} \qquad \underset{+_a, +_0}{=} \qquad \underset{S}{=} \tag{44}$$

$$\underset{+'}{=} \qquad \underset{+, S}{=} \qquad \underset{+_a}{=} \quad RHS \tag{45}$$

□
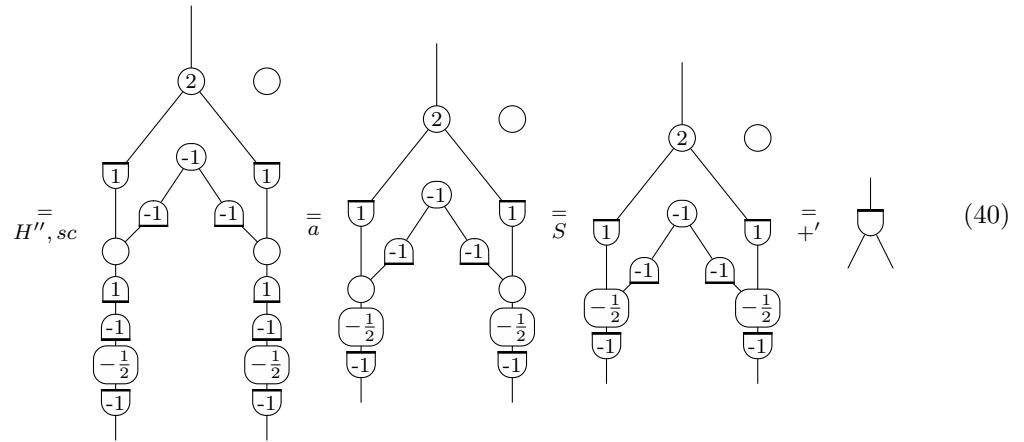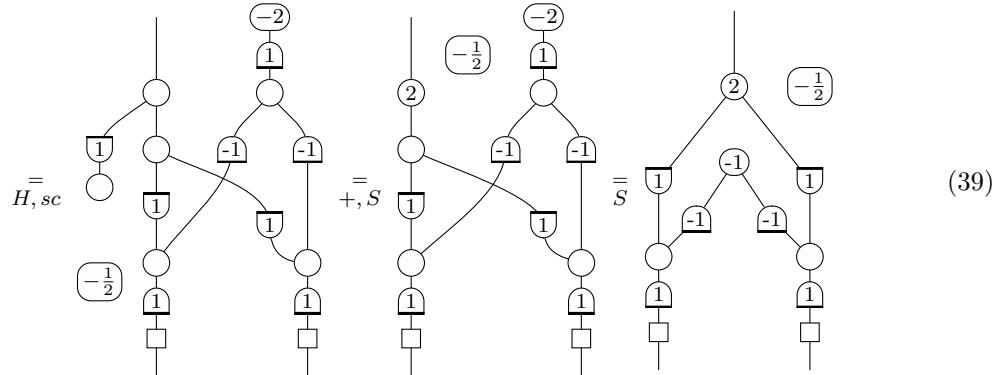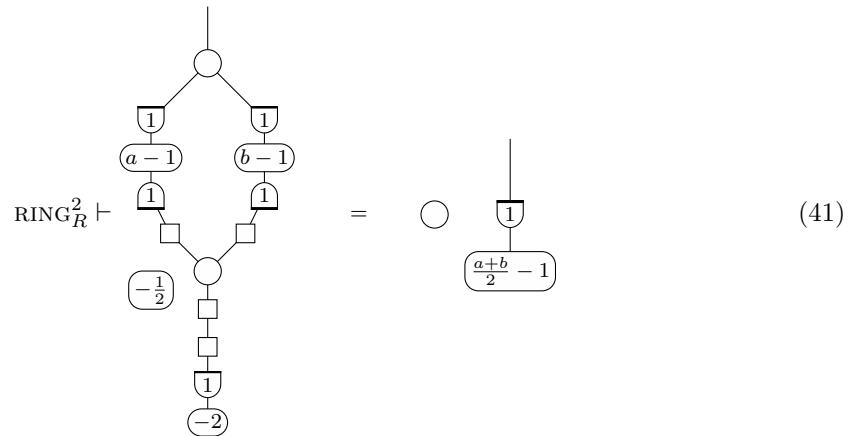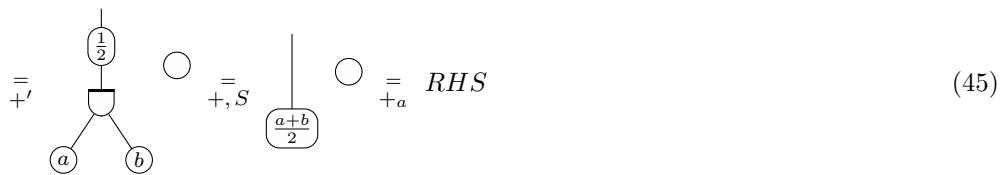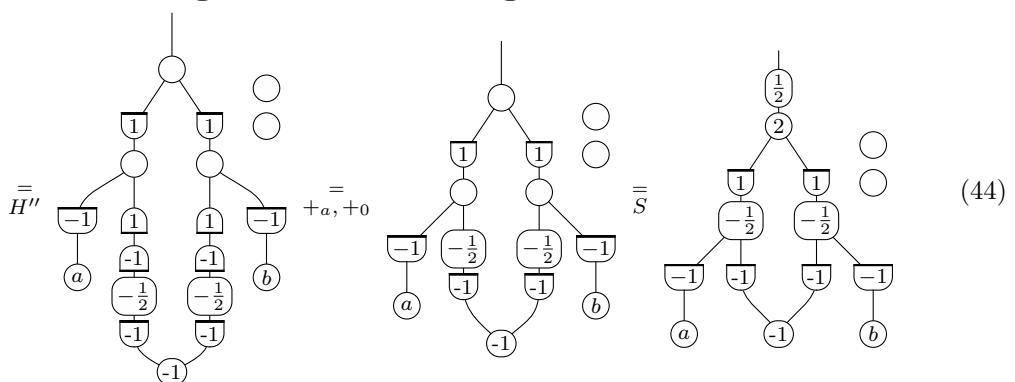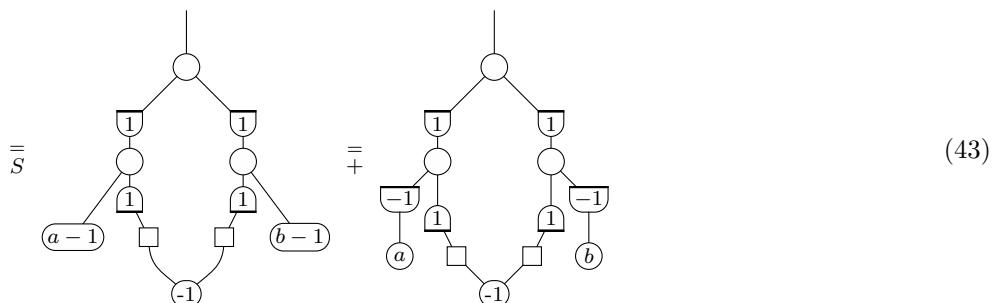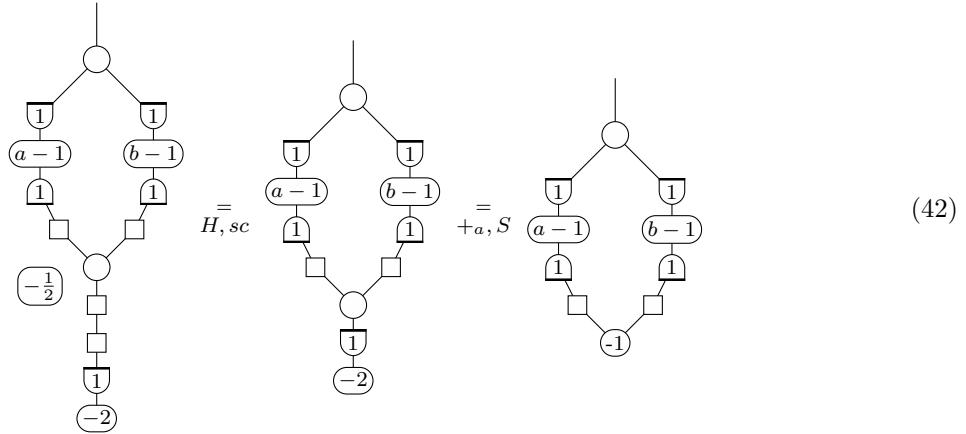
205

Lemma B.15. $\mathrm{RING}_R^2 \vdash F(ZH_I)$, *i.e.*

$$\mathrm{RING}_R^2 \vdash \quad = \quad \tag{46}$$

Proof.

$$\tag{47}$$

$$\tag{48}$$

$\square$

Lemma B.16. $\mathrm{RING}_R^2 \vdash F(ZH_O)$, *i.e.*

$$\tag{49}$$

Proof.

$$\tag{50}$$

$$\tag{51}$$

207

$$\overset{=}{\overset{=}{H''}} \qquad \overset{=}{\overset{=}{D}} \qquad \overset{=}{\overset{=}{D}} \tag{52}$$

$\square$

Lemma B.17. RING$_R^2$ $\vdash$ *equation* (16) *(the re-translation of the white spider) i.e.*

$$\text{RING}_R^2 \vdash \quad = \tag{53}$$

Proof. Nothing to prove $\qquad\qquad$ $\square$

Lemma B.18. RING$_R^2$ $\vdash$ *equation* (18) *(the re-translation of the state a) i.e.*

$$\text{RING}_R^2 \vdash \quad \underset{a}{\big|} \quad = \quad \underset{(a-1)\ (1)}{\bigvee} \tag{54}$$

Proof. Instance of the + rule $\qquad\qquad$ $\square$

We have shown that the ruleset RING$_R^2$ can prove all the translated rules of ZH$_R$, and can prove that generators equal their images under the translation into ZH and back. The ruleset RING$_R^2$ is therefore complete for R any commutative ring with a half, because ZH$_R$ is complete for any commutative ring with a half.

C Trigonometry proofs

In this appendix we prove the trigonometric proofs required in §6. We reproduce the side conditions for the rule ZX$_{EU'}$ for reference here:

In rule (EU'), β_1, β_2, β_3 and γ can be determined as follows: $x^+ := \frac{\alpha_1 + \alpha_2}{2}$, $x^- := x^- \alpha_2$, $z := -\sin(x^+) + i\cos(x^-)$ and $z' := \cos(x^+) - i\sin(x^-)$, then $\beta_1 = \arg z + \arg z'$, $\beta_2 = 2\arg(i + \frac{|z|}{|z'|})$, $\beta_3 = \arg z - \arg z'$, $\gamma = x^+ - \arg(z) + \frac{\pi - \beta_2}{2}$ where by convention $\arg(0) := 0$ and $z' = 0 \implies \beta_2 = 0$. (Figure 2, A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics [Vil19])

208

Lemma 6.43. *The quaternion* $\left(\pi, \frac{x+z}{\sqrt{2}}\right)$ *and its interactions with* (α, z) *and* (α, x):

$$\left(\pi, \frac{x+z}{\sqrt{2}}\right) \times (\alpha, z) = (\alpha, x) \times \left(\pi, \frac{x+z}{\sqrt{2}}\right) \tag{6.133}$$

$$(\alpha, z) \times \left(\pi, \frac{x+z}{\sqrt{2}}\right) = \left(\pi, \frac{x+z}{\sqrt{2}}\right) \times (\alpha, x) \tag{6.134}$$

$$\left(\pi, \frac{x+z}{\sqrt{2}}\right) \times \left(\pi, \frac{x+z}{\sqrt{2}}\right) = -1 \tag{6.135}$$

Proof.

$$(\pi, \frac{x+z}{\sqrt{2}}) \times (\alpha, z) = \frac{1}{\sqrt{2}}(i+k)(\cos\frac{\alpha}{2} + k\sin\frac{\alpha}{2}) \tag{55}$$

$$= \frac{1}{\sqrt{2}}(-\sin\frac{\alpha}{2} + i\cos\frac{\alpha}{2} - j\sin\frac{\alpha}{2} + k\cos\frac{\alpha}{2} \tag{56}$$

$$= \frac{1}{\sqrt{2}}(\cos\frac{\alpha}{2} + i\sin\frac{\alpha}{2})(i+k) \tag{57}$$

$$= (\alpha, x) \times (\pi, \frac{x+z}{\sqrt{2}}) \tag{58}$$

$$(\pi, \frac{x+z}{\sqrt{2}}) \times (\alpha, x) = \frac{1}{\sqrt{2}}(i+k)(\cos\frac{\alpha}{2} + i\sin\frac{\alpha}{2}) \tag{59}$$

$$= \frac{1}{\sqrt{2}}(-\sin\frac{\alpha}{2} + i\cos\frac{\alpha}{2} + j\sin\frac{\alpha}{2} + k\cos\frac{\alpha}{2} \tag{60}$$

$$= \frac{1}{\sqrt{2}}(\cos\frac{\alpha}{2} + k\sin\frac{\alpha}{2})(i+k) \tag{61}$$

$$= (\alpha, z) \times (\pi, \frac{x+z}{\sqrt{2}}) \tag{62}$$

$$(\pi, \frac{x+z}{\sqrt{2}}) \times (\pi, \frac{x+z}{\sqrt{2}}) = \frac{1}{\sqrt{2}}(i+k)\frac{1}{\sqrt{2}}(i+k) \tag{63}$$

$$= \frac{1}{2}(-1 - j - 1 + j) = -1 \tag{64}$$

$\square$

Lemma 6.44. *With the conditions of* $ZX_{EU'}$:

$$(\alpha_1, z) \times H \times (\alpha_2, z) = H \times (\beta_1, z) \times H \times (\beta_2, z) \times H \times (\beta_3, z) \times H \tag{6.136}$$

In the hope of easing legibility we separate out the real, i, j, and k components of quaternions onto separate lines where suitable.

Proof.

$$RHS = H \times (\beta_1, z) \times H \times (\beta_2, z) \times H \times (\beta_3, z) \times H \tag{65}$$

$$= H \times H \times H \times H \times (\beta_1, x) \times (\beta_2, z) \times (\beta_3, x) \tag{6.43} \tag{66}$$

209

$$=(\beta_1, x) \times (\beta_2, z) \times (\beta_3, x) \qquad (6.43) \qquad (67)$$

$$=(\cos\beta_1/2 + i\sin\beta_1/2)\times \qquad (68)$$
$$(\cos\beta_2/2 + k\sin\beta_2/2)\times$$
$$(\cos\beta_3/2 + i\sin\beta_3/2)$$

$$=1(\cos\beta_1/2\cos\beta_2/2\cos\beta_3/2 - \sin\beta_1/2\cos\beta_2/2\sin\beta_3/2)+ \qquad (69)$$
$$i(\cos\beta_1/2\cos\beta_2/2\sin\beta_3/2 + \sin\beta_1/2\cos\beta_2/2\cos\beta_3/2)+$$
$$j(\cos\beta_1/2\sin\beta_2/2\sin\beta_3/2 - \sin\beta_1/2\sin\beta_2/2\cos\beta_3/2)+$$
$$k(\cos\beta_1/2\sin\beta_2/2\cos\beta_3/2 + \sin\beta_1/2\sin\beta_2/2\sin\beta_3/2)$$

$$=1(\cos\beta_2/2)(\cos\beta_1/2\cos\beta_3/2 - \sin\beta_1/2\sin\beta_3/2)+ \qquad (70)$$
$$i(\cos\beta_2/2)(\cos\beta_1/2\sin\beta_3/2 + \sin\beta_1/2\cos\beta_3/2)+$$
$$j(\sin\beta_2/2)(\cos\beta_1/2\sin\beta_3/2 - \sin\beta_1/2\cos\beta_3/2)+$$
$$k(\sin\beta_2/2)(\cos\beta_1/2\cos\beta_3/2 + \sin\beta_1/2\sin\beta_3/2)$$

$$=1(\cos\beta_2/2)(\cos\frac{\beta_1+\beta_3}{2})+ \qquad (71)$$
$$i(\cos\beta_2/2)(\sin\frac{\beta_1+\beta_3}{2})+$$
$$j(\sin\beta_2/2)(\sin\frac{\beta_3-\beta_1}{2})+$$
$$k(\sin\beta_2/2)(\cos\frac{\beta_1-\beta_3}{2})$$

$$=1(\cos\beta_2/2)(\cos\arg z)+ \qquad (72)$$
$$i(\cos\beta_2/2)(\sin\arg z)+$$
$$j(\sin\beta_2/2)(-\sin\arg z')+$$
$$k(\sin\beta_2/2)(\cos\arg z')$$

$$\qquad (73)$$

Using properties of arguments and moduli we then show the following:

$$\cos\arg(a+ib) = a/|a+ib| \qquad (74)$$
$$\sin\arg(a+ib) = b/|a+ib| \qquad (75)$$
$$|z|^2 = \sin(x^+)^2 + \cos(x^-)^2 \qquad (76)$$
$$|z'|^2 = \sin(x^-)^2 + \cos(x^+)^2 \qquad (77)$$
$$|z|^2 + |z'|^2 = \cos^2 x^+ + \sin^2 x^+ + \cos^2 x^- + \sin^2 x^- = 2 \qquad (78)$$

$$\qquad (79)$$

$$\cos\arg z = \Re(z)/|z| = \frac{-\sin(\alpha_1+\alpha_2)/2}{|z|} \qquad (80)$$

$$\sin\arg z = \Im(z)/|z| = \frac{\cos(\alpha_1-\alpha_2)/2}{|z|} \qquad (81)$$

$$\cos\arg z' = \Re(z')/|z'| = \frac{\cos(\alpha_1+\alpha_2)/2}{|z'|} \qquad (82)$$

$$\sin \arg z' = \Im(z')/|z'| = \frac{-\sin(\alpha_1 - \alpha_2)/2}{|z'|} \tag{83}$$

$$\cos(\beta_2/2) = \cos \arg(i + |z|/|z'|) = \cos \arg(|z'|i + |z|) = |z|/\sqrt{2} \tag{84}$$

$$\sin(\beta_2/2) = \sin \arg(i + |z|/|z'|) = \sin \arg(|z'|i + |z|) = |z'|/\sqrt{2} \tag{85}$$

And now substitute these values into our expression for the right hand side:

$$
\begin{aligned}
RHS =& 1(\cos \beta_2/2)(\cos \arg z) + \\
& i(\cos \beta_2/2)(\sin \arg z) + \\
& j(\sin \beta_2/2)(-\sin \arg z') + \\
& k(\sin \beta_2/2)(\cos \arg z')
\end{aligned}
\tag{86}
$$

$$
\begin{aligned}
=& 1(|z|/\sqrt{2})(\frac{-\sin(\alpha_1 + \alpha_2)/2}{|z|}) + \\
& i(|z|/\sqrt{2})(\frac{\cos(\alpha_1 - \alpha_2)/2}{|z|}) + \\
& j(|z'|/\sqrt{2})(-\frac{-\sin(\alpha_1 - \alpha_2)/2}{|z'|}) + \\
& k(|z'|/\sqrt{2})(\frac{\cos(\alpha_1 + \alpha_2)/2}{|z'|})
\end{aligned}
\tag{87}
$$

$$
\begin{aligned}
=& (\frac{1}{\sqrt{2}}) \times \\
& (-1(\sin(\alpha_1 + \alpha_2)/2) + \\
& i(\cos(\alpha_1 - \alpha_2)/2) + \\
& j(\sin(\alpha_1 - \alpha_2)/2) + \\
& k(\cos(\alpha_1 + \alpha_2)/2))
\end{aligned}
\tag{88}
$$

$$\tag{89}$$

And now for the left hand side:

$$LHS = (\alpha_1, z) \times H \times (\alpha_2, z) \tag{90}$$

$$= \frac{1}{\sqrt{2}}(\cos \alpha_1 + k \sin \alpha_1)(i + k)(\cos \alpha_2 + k \sin \alpha_2) \tag{91}$$

$$= \frac{1}{\sqrt{2}}(i \cos \alpha_1 \cos \alpha_2 - j \cos \alpha_1 \sin \alpha_2 + k \cos \alpha_1 \cos \alpha_2 - \cos \alpha_1 \sin \alpha_2 + \tag{92}$$

$$j \sin \alpha_1 \cos \alpha_2 + i \sin \alpha_1 \sin \alpha_2 - \sin \alpha_1 \cos \alpha_2 - k \sin \alpha_1 \sin \alpha_2)$$

$$= \frac{1}{\sqrt{2}}(-(\cos \alpha_1 \sin \alpha_2 + \sin \alpha_1 \cos \alpha_2) \tag{93}$$

$$i(\sin \alpha_1 \sin \alpha_2 + \cos \alpha_1 \alpha_2) +$$

$$j(\sin \alpha_1 \cos \alpha_2 - \cos \alpha_1 \sin \alpha_2) +$$

$$k(\cos \alpha_1 \cos \alpha_2 - \sin \alpha_1 \sin \alpha_2))$$

$$= (\frac{1}{\sqrt{2}}) \times \tag{94}$$

$$(-1(\sin(\alpha_1 + \alpha_2)/2)+$$
$$i(\cos(\alpha_1 - \alpha_2)/2)+$$
$$j(\sin(\alpha_1 - \alpha_2)/2)+$$
$$k(\cos(\alpha_1 + \alpha_2)/2))$$

$\square$

Proposition C.1. *The map ϕ, given by*

$$\phi : \qquad (\textit{Unit Quaternions}, \times) \to (2 \times 2 \textit{ complex matrices}, \circ) \tag{95}$$

$$\phi : \qquad q_w + iq_x + jq_y + kq_z \mapsto \begin{pmatrix} q_w - iq_z & q_y - iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix} \tag{96}$$

is a group homomorphism with trivial kernel.

Proof. Write q_1 as $w + x + y + z$ and q_2 as $w' + x' + y' + z'$:

- Show that $\phi(1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$:

$$\phi(1) = \begin{pmatrix} 1 - i0 & 0 - i0 \\ -0 - i0 & 1 + i0 \end{pmatrix} \tag{97}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{98}$$

- Show that $\phi(q_1)\phi(q_2) = \phi(q_1 \times q_2)$:

$$LHS = \begin{pmatrix} w - iz & y - ix \\ -y - ix & w + iz \end{pmatrix} \circ \begin{pmatrix} w' - iz' & y' - ix' \\ -y' - ix' & w' + iz' \end{pmatrix} \tag{99}$$

$$LHS_{(1,1)} = ((ww' - zz' - yy' - xx') - i(wz' - xy' + yx' + w'z)) \tag{100}$$

$$LHS_{(1,2)} = (((w - iz)(y' - ix') + (y - ix)(w' + iz')) \tag{101}$$

$$= (wy' - zx' + yw' + xz') - i(wx' + xw' - yz' + zy') \tag{102}$$

$$LHS_{(2,1)} = ((-y - ix)(w' - iz') + (w + iz)(-y' - ix')) \tag{103}$$

$$= -(yw' + xz' + wy' - zx') - i(wx' + xw' - yz' + zy') \tag{104}$$

$$LHS_{(2,2)} = ((-y - ix)(y' - ix') + (w + iz)(w' + iz')) \tag{105}$$

$$= (ww' - xx' - yy' - zz') + i(wz' - xy' + yx' + zw') \tag{106}$$

$$RHS_{(1,1)} = (ww' - xx' - yy' - zz') - i(wz' + xy' - yx' + zw') \tag{107}$$

$$RHS_{(1,2)} = (wy' - xz' + yw') + zx' - i(wx' + xw' + yz' - zy') \tag{108}$$

$$RHS_{(2,1)} = -(wy' - xz' + yw') + zx') - i(wx' + xw' + yz' - zy') \tag{109}$$

$$RHS_{(2,2)} = (ww' - xx' - yy' - zz') + i(wz' + xy' - yx' + zw') \tag{110}$$

- Show that $\phi(q) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \implies q = 1$: Looking at the matrix entries individually:

$$1 = q_w - iq_z \tag{111}$$
$$\implies q_w = 1 \tag{112}$$
$$\implies q_z = 0 \tag{113}$$
$$0 = -q_y - iq_x \tag{114}$$
$$\implies q_y = 0 \tag{115}$$
$$\implies q_x = 0 \tag{116}$$
$$\therefore q = 1 \tag{117}$$

$\square$

D   Proof of the n-joined ZH spider law

Lemma 7.4 (n-joined ZH spider law). *In the ZH calculus:*

$$a, b \in \mathbb{C} \setminus \{1\}$$
$$n > 0 \tag{7.4}$$

Proof. Aiming to proceed by induction we first note the base case where $n = 1$:

$$\tag{118}$$

This is shown by checking soundness by hand. For the induction step we will need the following lemma:

$$\tag{119}$$

Which again can be shown to be sound by direct computation. For the induction step we assume the original statement is true for $n-1$, $n > 1$, we then add another joining Hadamard.

213

$$1 + (\tfrac{-1}{2})^{n-1}(b-1)(a-1) \qquad (120)$$

$$(3 - 1 - (\tfrac{-1}{2})^{n-1}(b-1)(a-1))/2 \;=\; 1 + (\tfrac{-1}{2})^{n}(b-1)(a-1) \qquad (121)$$

$\square$

E Worked example of !-box removal

In this appendix we give a longer example of !-box removal. Consider the following $\mathrm{ZH}_{\mathbb{C}}$ equation, which is not sound, but is suitable for demonstrating the principle of !-box removal.

$$\qquad (122)$$

We first need to work out the *join*, which we do by counting the wires entering the !-boxes.

$$2^1 + 2^0 = 3 \qquad (123)$$

By Theorem 8.51 we need to check $d = \{0, 1, 2, 3\}$ instances of the !-box. So in order to verify:

$$\qquad (124)$$

We just need to verify these equations:

214

$$(125)$$

$$(126)$$

Note that these equations contain no !-boxes.